

O

AR-009-739

DSTO-RR-0086

F

Common Security Protocol Security
Labelling and its Applications

M. K. F. Lai, J. Burgess, K. Forrest,
H. Daniel and N. F. Parker

S

DISPOSITION STATEMENT E
Approved for public release
Distribution: Unrestricted

APPROVED FOR PUBLIC RELEASE

© Commonwealth of Australia

19961009 130

I

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

UNCLASSIFIED

Common Security Protocol Security Labelling and its Applications

M.K.F. Lai, J. Burgess, K. Forrest,
H. Daniel & N.F. Parker

**Information Technology Division
Electronics and Surveillance Research Laboratory**

DSTO-RR-0086

ABSTRACT

This paper is part of the documentation series produced under the HQADF sponsored task "D6: A Security Architecture for Large, Distributed Multimedia Systems". It shows that the functionality of the Defence adopted Common Security Protocol currently is insufficient to realise an electronic analogue to the paper based formal military correspondence composition. Some minimal structural changes to the protocol data unit and the corresponding procedural changes to the protocol handling are proposed to address the deficiency.

DTIC QUALITY INSPECTED

APPROVED FOR PUBLIC RELEASE

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

UNCLASSIFIED

UNCLASSIFIED

Published by

*DSTO Electronics and Surveillance Research Laboratory
PO Box 1500
Salisbury, South Australia, 5108*

*Telephone: (08) 259 7053
Fax: (08) 259 5619*

*© Commonwealth of Australia 1996
AR No. 009-739
July 1996*

APPROVED FOR PUBLIC RELEASE

UNCLASSIFIED

Common Security Protocol Security Labelling and its Applications

Executive Summary

With the exception of voice, messaging is the most common form of communications. In the day-to-day command and control activities of Defence, military messages and their specialised types (such as formal military correspondences) are indispensable at all levels. Their roles are detailed explicitly in Defence doctrines. Military messaging also provides certain technical advantages over voice. Messaging operates in a store and forward mode, where an end-to-end connection between originator and recipient is not required before a message is sent. Being connectionless, messaging can be supported by "writer-to-reader" security.

The digital nature of messaging allows security to be applied to a message (once only) at its origin, and that security to be preserved unaltered between originator and recipient. The control of message security resides with each originator and recipient, and is described as writer-to-reader security. All messages are secured by the originators through a uniform set of originator-mechanisms. Similarly, secured messages are accessed by their intended recipients through a uniform set of recipient-mechanisms. Security policies can be implemented via these originator and recipient mechanisms, and conformance with these security policies is independent of operational scale.

Writer-to-reader security contrasts with the traditional security techniques where security is provided by the communications infrastructure (typically at the link layer) and by the computer infrastructure (typically at the operating system level). Writer-to-reader security aims to minimise:

- the degree of trust in the communications and computer infrastructures in terms of their security requirement; and
- the requirement for red¹ gateways between systems of different security levels (e.g. between classified and unclassified systems) or different operational environments (e.g. between fixed strategic and mobile tactical systems).

Common Security Protocol (CSP) or its technical equivalent Message Security Protocol (MSP) has been designed specifically to enable the writer-to-reader security for military messaging, particularly that based on international standards. The crucial function of CSP is its encapsulation of a message within its protocol data unit structure (namely inside its **encapsulatedContent**). Information security (infosec) services such as:

- message confidentiality;
- non-repudiation with proof of message origin authentication;
- non-repudiation with proof of message delivery; and
- message submission or delivery access control

are provided to the encapsulated message through the appropriate utilisation of the basic CSP infosec protocol mechanisms such as:

1. In the context of gateways, "red" refers to transmission (e.g. protocol conversion) that is processed in clear, whereas "black" refers to transmission that is processed encrypted.

UNCLASSIFIED

- exchange of message encryption keys for authorised access to encrypted messages (i.e. via its **recipientSecurityData**);
- generation of an originator's signature associated with the encapsulated message (i.e. via the **signatureInformation** and **signatureValue** belonging to its **signatureBlock**);
- generation of a recipient's signature associated with a message receipt (i.e. via the **receiptInformation** and **signatureValue** belonging to its **signatureBlock**); and
- security labelling of the encapsulated message (i.e. via the **securityLabel** belonging to the **messageSecurityData** inside its **originatorSecurityData**).

This paper aims to explore the application of CSP based infosec to the more general case of the command and control information item than the basic military message. One specific class of information items of interest is the formal military correspondence such as commander/minister/secretary minutes or letters, command and control directives, and military operational orders or plans. This class generally has more security requirements. Formal military correspondence typically consists of a primary part and a number of secondary parts as annexes or enclosures. Each of these annexes or enclosures may have originated from, someone other than the originator of the current correspondence. In addition, an annex or enclosure could be assigned its own unique security classification (or, more generally, security label) which may be different from that of the overall military correspondence. In some cases, a military correspondence may also need to be treated as an accountable document (which imposes further restraints in terms of the security requirements).

In an attempt to apply the CSP based writer-to-reader security to the composition of the more general formal military correspondence, the CSP-based message forwarding function presents a promising approach for the following reasons:

- the primary and each secondary part of a formal military correspondence could be considered as a forwarded CSP protocol data unit which encapsulates that part; and
- the overall formal military correspondence could be considered as a military message which includes all the forwarded CSP protocol data units.

This calls for an examination of the CSP-based message forwarding function and its associated infosec protocol mechanisms. Specifically, this paper shows that the CSP-based forwarding function (as currently defined in CSP or MSP) is not sufficient to provide an adequate electronic analogue to the paper-based formal military correspondence. Consequently, this paper proposes some minimal changes to the CSP protocol data unit structure, and corresponding changes to the CSP handling procedures. The structural changes mainly involve the relocation of the **messageSecurityData** from the **originatorSecurityData** to the **signatureBlock**, while the procedural changes require that the generation of an originator's signature (associated with the encapsulated message) also takes the associated **messageSecurityData** into account. Based on the proposed changes, this paper demonstrates a better analogue of the paper-based formal military correspondence.

UNCLASSIFIED

Authors

Lai M.K.F.

Information Technology Division

Dr Lai received his B.Sc. (Mathematical Science) First Class Honours degree in 1984, followed by his Ph.D. in Combinatorial Group Theory completed in 1987, both from the University of London. He is currently a Senior Research Scientist in the Information Security Section of Trusted Computer Systems Group at DSTO.

Neville F Parker

Communications Division

Dr Parker received a PhD degree from The University of Adelaide in 1979 and subsequently obtained equivalent to a First Class Honours degree in Computer Science from Flinders University of South Australia. He has worked in industry as a software engineer and consultant. Since joining DSTO he has been working in information security, and communications in tactical environments, with a particular interest in military messaging.

Helen A Daniel

Information Technology Division

Mrs Daniel received a B.Sc (mathematics) from Monash University in 1988. She is currently a Information and Technology Officer in the Trusted Computer Systems Group at DSTO.

UNCLASSIFIED

UNCLASSIFIED

John Burgess

Department of Defence

Mr John Burgess is employed by the Australian National Information Security Authority and is currently involved in development of Information Security Policy for military messaging and electronic key management systems.

Wing Commander Kim Forrest

Defence Materiel Division

Wing Commander Forrest joined the RAAF in 1977 and undertook a Bachelor of Engineering in Communications and Electronics at the Royal Melbourne Institute of Technology. In 1984, he completed the Advanced Systems Engineering Course at the Royal Air Force College, Cranwell, UK, leading to the award of a Master of Science in Aerosystems Engineering from the Loughborough University of Technology. Since September 1994, he has been the Project Manager of the Defence Messaging and Directory Service project.

UNCLASSIFIED

Contents

1	INTRODUCTION	1
2	MESSAGE SUBMISSION BASED ON ACP123 AND ACP120	3
2.1	<i>Electronic Military Message Composition</i>	3
2.1.1	<i>An Analogue of Paper Based Military Correspondence Composition</i>	4
2.2	<i>Security Service Selection</i>	5
2.3	<i>CSP PDU Construction</i>	5
2.3.1	<i>Confidentiality Provision</i>	6
2.3.2	<i>Digital Signature Creation</i>	6
2.3.3	<i>Message Security Data Inclusion</i>	7
2.4	<i>Functions Required at the Submission Port</i>	8
2.5	<i>CSP PDU Forwarding</i>	8
2.5.1	<i>Alternative States of Forwarded CSP PDUs</i>	8
2.5.1.1	<i>Encrypted State</i>	8
2.5.1.2	<i>Clear Text State</i>	9
2.5.2	<i>CSP PDU Forwarding Procedures</i>	11
2.5.3	<i>An Application of the Forwarding Mechanisms</i>	12
2.5.4	<i>Some Security Weaknesses Associated with CSP PDU Forwarding</i>	14
3	PERMANENT INTEGRITY-GUARANTEED SECURITY LABEL	17
3.1	<i>Proposed CSP PDU Structural Changes</i>	17
3.1.1	<i>Changes in the OriginatorSecurityData Sub-Structure</i>	17
3.1.2	<i>Changes in the SignatureBlock Sub-Structure</i>	18
3.1.3	<i>Changes to Forwarded and Retained CSP PDUs</i>	20
3.2	<i>Proposed Changes to the CSP PDU Handling Procedures</i>	21
3.2.1	<i>Changes To Message Submission</i>	21
3.2.1.1	<i>Consequences of MessageSecurityData Activation</i>	22
3.2.1.2	<i>CSP Submission Access Control Determination</i>	23
3.2.1.3	<i>Implied Changes to Signature Generation</i>	23
3.2.2	<i>Changes to Message Reception and Forwarding</i>	25
3.3	<i>A Closer Electronic Analogue to Paper Based Formal Military Correspondence Composition</i>	27
3.4	<i>Consideration of Other Approaches</i>	29
4	CONCLUSION	31
5	REFERENCES	33
	DISTRIBUTION	35

UNCLASSIFIED

Figures

- FIGURE 1 THE STRUCTURE OF A P772 PDU OF ACP123 3
- FIGURE 2 A TYPICAL PAPER BASED FORMAL MILITARY
CORRESPONDENCE 4
- FIGURE 3 MESSAGE SUBMISSION BASED ON ACP123 AND
ACP120 5
- FIGURE 4 THE STRUCTURE AND SEMANTIC OF A CSP PDU 6
- FIGURE 5 A FORWARDED CSP PDU IN THE ENCRYPTED STATE
9
- FIGURE 6 A FORWARDED OR RETAINED CSP PDU IN THE
CLEAR TEXT STATE 10
- FIGURE 7 THE ACTION SEQUENCE ASSOCIATED WITH MM
FORWARDING 11
- FIGURE 8 ELECTRONIC ANALOGUE TO A PAPER BASED
MILITARY CORRESPONDENCE 13
- FIGURE 9 PROPOSED CHANGES TO THE
ORIGINATORSECURITYDATA SUB-
STRUCTURE 17
- FIGURE 10 PROPOSED CHANGES TO THE SIGNATUREBLOCK
SUB-STRUCTURE 19
- FIGURE 11 IMPLIED CHANGES TO A FORWARDED CSP PDU IN
THE ENCRYPTED STATE 20
- FIGURE 12 IMPLIED CHANGES TO FORWARDED OR RETAINED
CSP PDU IN THE CLEAR TEXT STATE 21
- FIGURE 13 IMPLIED CHANGES TO HASH CALCULATION &
SIGNATURE GENERATION 24
- FIGURE 14 RELATIONSHIPS BETWEEN OBJECTS OF A
FORWARDED CSP PDU IN THE CLEAR FORM
25
- FIGURE 15 CHANGES TO THE ACTION SEQUENCE ASSOCIATED
WITH MM FORWARDING 26
- FIGURE 16 IMPROVED ELECTRONIC ANALOGUE TO A PAPER
BASED MILITARY CORRESPONDENCE 28

UNCLASSIFIED

1 Introduction

With the exception of voice, messaging is the most common form of communications. In the day-to-day command and control activities of Defence, military messages and their specialised types (such as formal military correspondences) are indispensable at all levels. Their roles are detailed explicitly in Defence doctrines [1], [2], [3] & [4]. Military messaging also provides certain technical advantages over voice. Messaging operates in a store and forward mode where an end-to-end connection between originator and recipient is not required before a message is sent. Being connectionless, messaging can be supported by "writer-to-reader" security.

The digital nature of messaging allows security to be applied to a message (once only) at its origin, and that security to be preserved unaltered between originator and recipient. The control of message security resides with each originator and recipient, and is described as writer-to-reader security. All messages are secured by the originators through a uniform set of originator-mechanisms. Similarly, secured messages are accessed by their intended recipients through a uniform set of recipient-mechanisms. Security policies can be implemented via these originator and recipient mechanisms, and conformance with these security policies is independent of operational scale.

Writer-to-reader security contrasts with the traditional security techniques where security is provided by the communications infrastructure (typically at the link layer) and by the computer infrastructure (typically at the operating system level). Writer-to-reader security aims to minimise:

- the degree of trust in the communications and computer infrastructures in terms of their security provision requirement; and
- the requirement for red¹ gateways between systems of different security levels (e.g. between classified and unclassified systems) or different operational environments (e.g. between fixed strategic and mobile tactical systems).

Common Security Protocol (CSP) [6] or its technical equivalent Message Security Protocol (MSP) [7] has been designed specifically to enable the writer-to-reader security for military messaging, particularly that based on [8] and [9]. The crucial function of CSP is its encapsulation of a message within its protocol data unit (PDU) structure (namely inside its **encapsulatedContent**). Information security (infosec) services such as:

- message confidentiality;
- non-repudiation with proof of message origin authentication;
- non-repudiation with proof of message delivery; and
- message submission or delivery access control

are provided to the encapsulated message through the appropriate utilisation of the basic CSP infosec protocol mechanisms such as

- exchange of message encryption keys for authorised access to encrypted messages (i.e. via its **recipientSecurityData**);
- generation of an originator's signature associated with the encapsulated message (i.e. via the **signatureInformation** and **signatureValue** belonging to its **signatureBlock**);
- generation of a recipient's signature associated with a message receipt (i.e. via the **receiptInformation** and **signatureValue** belonging to its **signatureBlock**); and
- security labelling of the encapsulated message (i.e. via the **securityLabel** belonging

1. In the context of gateways, "red" refers to transmission (e.g. protocol conversion) that is processed in clear, whereas "black" refers to transmission that is processed encrypted.

to the **messageSecurityData** inside its **originatorSecurityData**).

This paper aims to explore the application of CSP based infosec to a more general case of command and control information item than the basic military message. One specific class of information item of interest is the formal military correspondence such as commander/minister/secretary minutes or letters, command and control directives, and military operational orders or plans. This class generally has more security requirements. A formal military correspondence typically consists of a primary part and a number of secondary parts as annexes or enclosures. Each of these annexes or enclosures may be written by, or have originated from, someone other than the originator of the current correspondence. In addition, an annex or enclosure could be assigned its own unique security classification (or, more generally, security label) which may be different from that of the overall military correspondence. In some cases, a military correspondence may also need to be treated as an accountable document (which imposes further restraints in terms of the security requirements).

In an attempt to apply the CSP based writer-to-reader security to the composition of the more general formal military correspondence, the CSP-based message forwarding function presents a promising approach for the following reasons:

- the primary and each secondary part of a formal military correspondence could be considered as a forwarded CSP PDU which encapsulates that part; and
- the overall formal military correspondence then could be considered as a military message which includes all the forwarded CSP PDUs.

This calls for an examination of the CSP-based message forwarding function and its associated infosec protocol mechanisms. However, as it will be shown in Section 2, the CSP-based forwarding function (as currently defined in [6] or [7]) is not sufficient to provide an adequate electronic analogue to the paper-based formal military correspondence composition. Consequently, this paper proposes some minimal changes to the CSP PDU structure and corresponding changes to the CSP handling procedures (Section 3). The structural changes mainly involve the relocation of the **message security data** from the **originatorSecurityData** to the **signatureBlock**, while the procedural changes require that the generation of an originator's signature (associated with the encapsulated message) also takes the associated **messageSecurityData** into account. Finally, based on the proposed changes, Section 3 demonstrates a more appropriate electronic analogue of the paper-based formal military correspondence.

2 Message Submission Based on ACP123 and ACP120

In this section, we briefly explain the main processes of message submission based on ACP123 [8] and ACP120 [6].

According to the messaging paradigm of X.400 [9], every (authorised) message user should be able to access a user agent (UA). This UA is the user interface to the messaging system, connected to message transfer agents (MTAs) possibly through a message store (MS). The connected MTAs comprise the message transfer system (MTS), which is considered as an unprotected message transportation infrastructure. The UA facilitates the user requirements of composing as well as reading messages (which are formatted in the X.420 interpersonal message (IPM) structure [10]).

2.1 Electronic Military Message Composition

The Defence-endorsed ACP123 standards have defined the military extensions to the X.420 IPM structure. The UA, called an ACP123 UA, must be able to handle these military extensions. A military message (MM) is therefore an IPM formatted message with the ACP123 military extensions. The set of protocol procedures for processing MMs defines the semantics of MM handling between an originator's UA and a recipient's UA. This knowledge is not needed by any other entities of the MTS, and ACP123 military messaging can therefore be called "writer-to-reader". A MM also may be referred as a P772 PDU.

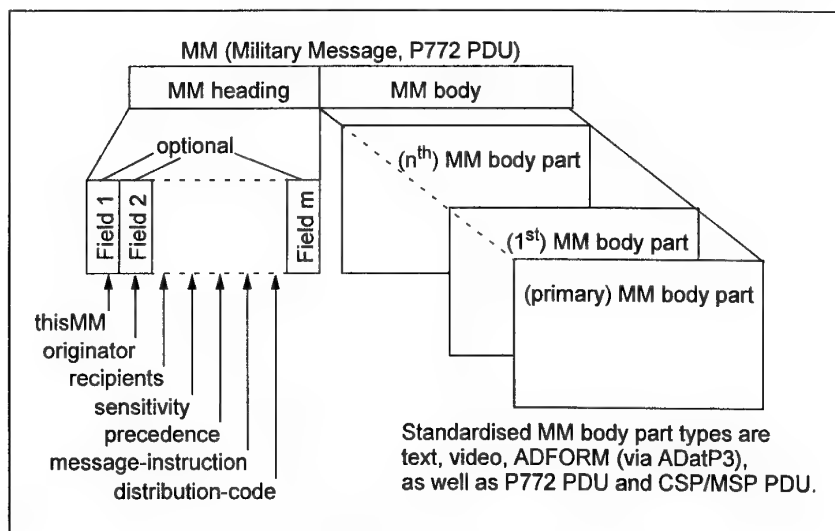


Figure 1 The Structure of a P772 PDU of ACP123

The structure of the P772 PDU is presented in Figure 1. In particular, it allows multiple body parts of different types within a single MM. A number of body part types have been defined in X.420 and ACP123. These include IA5-text, fax, voice, ADatP3, and forwarded MM. Externally defined body part types are also allowed. In Section 2.5, we will focus on a particular externally defined body part type known as **Forwarded-CSP-Message-Body-Part**. It will be seen that this body part is used to include an "integrity-guaranteed" information item which has been created previously, or received from a third party. In this paper, an information item is said to be integrity-guaranteed only if its content cannot be altered accidentally or intentionally without detection.

2.1.1 An Analogue of Paper Based Military Correspondence Composition

The composition of a MM with multiple body parts of different types is intended to be an electronic analogue of the traditional paper based formal military correspondence. In a typical paper based formal military correspondence, the substance of the communication is presented in a (covering) minute, letter, directive, order, operational plan, or equivalents. Additional secondary information and materials (such as graphics, maps, demographic data, intelligence reports or summaries, soldier's handbooks, weapons recognition guides, and press releases) are normally appended as annexes, or enclosures. The content of an annex or enclosure may have been written by, or originated from, someone other than the originator of the current correspondence. Moreover, this content may have its own security classification (or more generally its own security label) assigned by its writer or originator. The overall security classification (or label) of the formal correspondence, therefore, must be dominated by the most restrictive classification (or label) among the primary material and the secondary enclosures/annexes.

There are numerous examples of formal military correspondence which are indispensable in the current day-to-day command and control activities of Defence at all levels. The role of this correspondence is explained in various formal Defence doctrines such as [1], [2], [3] & [4]. A typical paper based formal military correspondence is depicted in Figure 2.

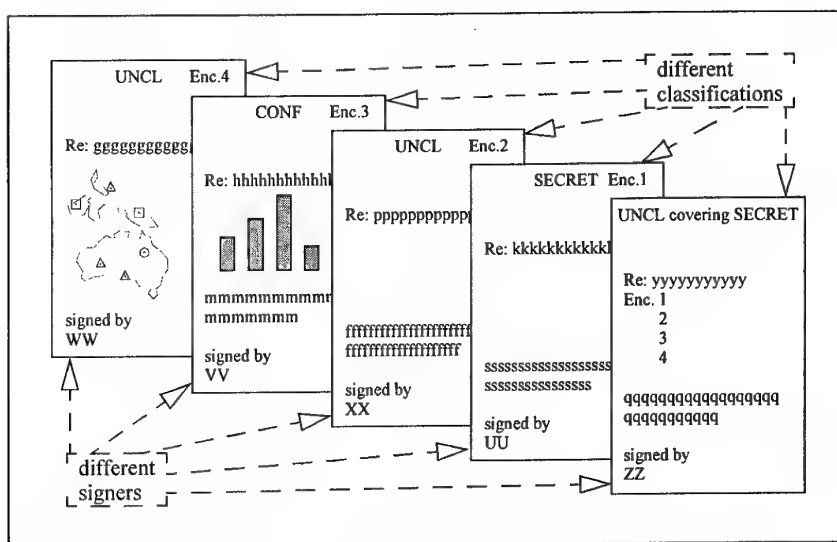


Figure 2 A Typical Paper Based Formal Military Correspondence

In some cases, a formal military correspondence also has to be treated as an accountable document which is to be recorded in the recipient's classified document register, and is to be mustered and reported to the originator's organisation at fixed intervals [2]. It may be required that the accountable document is not to be copied, destroyed or passed to another branch/division/directorate without the prior approval of the originator's organisation. Moreover, every copy of an accountable document is required to have a unique copy number. Every recipient on the distribution list should receive only his/her designated copy of the accountable document. This copy must be identifiable via the unique copy number associated with his/her name/position on the distribution list. In the US there also are similar requirements in connection with the principles for classifying, safeguarding, and declassifying national security information [5].

In Section 2.5.3, we shall discuss an approach to achieving an electronic analogue to the paper based formal military correspondence system. This approach will rely on the

(information) security services selected for protecting the electronic MMs. The selection of these security services is discussed in the next subsection.

2.2 Security Service Selection

While composing a MM (P772 PDU) with an ACP123 UA, the originator may also select a range of basic information security (infosec) services to protect the MM on a per-message basis. These basic security services are message confidentiality, message origin authentication, message integrity, message access control, non-repudiation with proof of origin, and non-repudiation with proof of delivery. The protocol mechanism associated with these security services is the Common Security Protocol (CSP, as specified in ACP120 [6]) or the Message Security Protocol (MSP, specified in [7]). While the MSP development is controlled and managed by US, the CSP development is controlled and managed by the CCEB (US, UK, Canada, NZ, and Australia). Technically, there should be no significant difference between MSP and CSP that would impact on interoperability. For the purpose of this paper, the terms MSP and CSP are interchangeable.

The originator is provided a CSP front end from which the desired security services are selected for the MM being composed. This results in a set of selected security service indicators. The completed MM and its associated set of selected security service indicators are then passed to the CSP process unit. This passage is shown in Figure 3. Based on the selected security service indicators, the CSP process unit constructs an appropriate CSP PDU (denoted by **Csp** in Figure 4). This CSP PDU protects the MM (P772 PDU) by encapsulating it inside its PDU structure. It is evident that the combined functionality of the CSP front end and CSP process unit is identical to the functionality of the CSP user agent explained in [6] & [7].

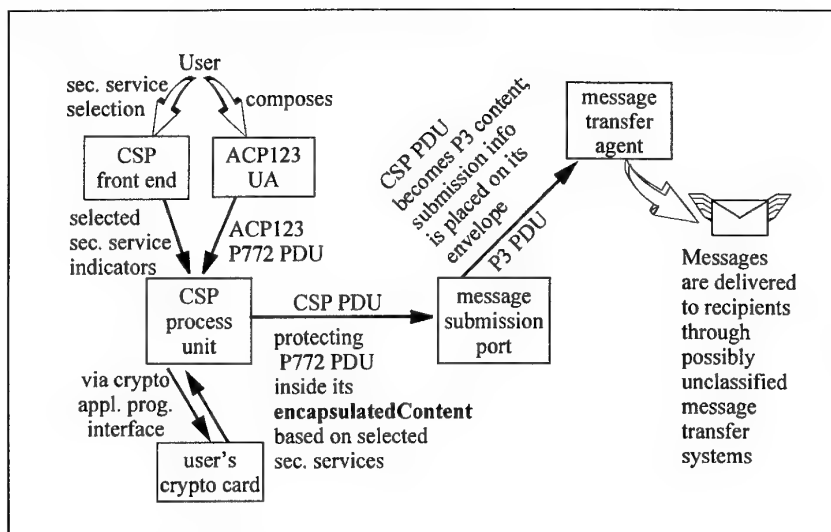


Figure 3 Message Submission Based on ACP123 and ACP120

2.3 CSP PDU Construction

The generic structure of a CSP PDU is depicted in Figure 4. As an information object, the CSP PDU is denoted by **Csp**. The originator's MM (P772 PDU) is encapsulated inside the **encapsulatedContent** field. Through various mechanisms of the MTS, the constructed CSP PDU is passed from the originator to the intended recipients without alternation from any intervening MTS entities. Hence, the security services that the CSP PDU enables are called

“writer-to-reader” security. This is only possible because P772 is itself a “writer-to-reader” protocol, as explained in Section 2.1.

2.3.1 Confidentiality Provision

If the originator has selected the confidentiality service, then it is the encrypted form of the MM that is placed inside the **encapsulatedContent** field. Otherwise, it suffices to place the MM in its clear text form inside this field. The encryption of the MM is handled by the originator’s crypto card which has the ability to generate an appropriate message encryption key (**msgKey**). The crypto card also contains the originator’s private key materials specific to a given key exchange (or management) arrangement. The knowledge of this **msgKey** has to be conveyed to the intended recipients securely so that the clear text MM can be recovered by only those intended recipients. The **msgKey**, therefore, is hidden cryptographically inside the **recipientSecurityData** of the CSP PDU (Figure 4) via the key exchange (or management) arrangement between the originator and the intended recipients. The CSP process unit communicates with the crypto card via the crypto application program interface (CAPI) to request the card to perform various cryptography-specific activities (Figure 3).

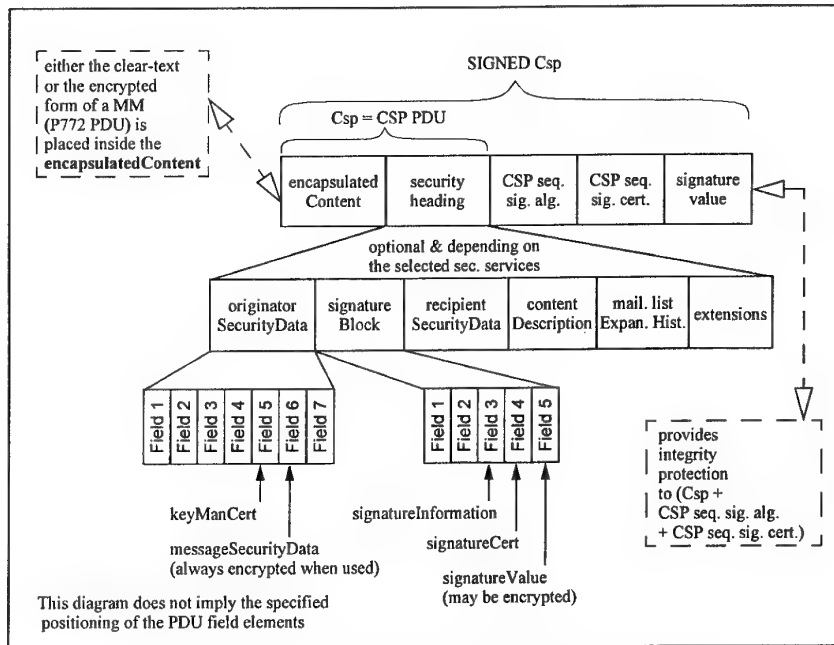


Figure 4 The Structure and Semantic of a CSP PDU

2.3.2 Digital Signature Creation

If the originator has requested the security service of non-repudiation with proof of origin, then a digital signature (denoted by **signatureValue**) associated with the MM needs to be created and subsequently placed inside the **signatureBlock** field of the CSP PDU (Figure 4). If the MM is to be encrypted, then so should the **signatureValue**. As with the MM encryption, the MM signature creation is handled by the originator’s crypto card.

During the signature creation, the crypto card uses the originator’s private signature key materials (originally installed inside the card when issued to the originator by the issuing authority). According to [6], it is not only the MM (P772 PDU) that is signed, but it is the combination of the P772 PDU and the **signatureInformation** that is signed. The **signatureInformation** is placed inside the **signatureBlock** of the CSP PDU (Figure 4). It consists of four components and they are explained as follows:

- **CspContentType** (optional)
It specifies the content type of the information object placed inside the CSP **encapsulatedContent**. Within the context of this paper, it is **id-nato-mmhs-cont-mm88** (which refers to the MM content type [11]).
- **signedContentIdentifier**
It should be identical to the **MM-id** (which may be viewed as a reference number) generated by the originator and placed inside the **thisMM** field of MM **heading** (Figure 1) of the MM being composed.
- **receiptRequests**
It indicates from which recipients the originator requests signed receipts.
- **receiptsTo** (optional)
It identifies the entities to whom the recipients should send signed receipts.

Hence, the signature not only guarantees the integrity of the originator's MM, but also the integrity of the **signatureInformation** (associated with the MM) assigned by the originator. To verify the signature associated with a MM, a recipient must use the associated **signatureInformation**. This implies that the recipient cannot ignore the receipt-instruction demanded as set out by the originator in the **receiptRequests** and **receiptsTo** subfields of the **signatureInformation**.

2.3.3 Message Security Data Inclusion

The originator may optionally select the inclusion of the **messageSecurityData** (associated with his/her MM) inside the **originatorSecurityData** field (Figure 4). If the **messageSecurityData** is provided inside the CSP PDU, then it has to be protected via an encryption with the **msgKey** ([6] or [7]). In other words, the **messageSecurityData** does not appear in its clear text form during the CSP PDU transportation. By the definition of **msgKey**, it also follows that the CSP PDU **encapsulatedContent** should contain the encrypted form of the MM instead of the MM in its clear text form. Consequently, the **signatureValue** would have to be encrypted also if a digital signature has been requested by the originator (as in accordance with [6] or [7]).

The **messageSecurityData** consists of the **securityLabel** and **kmAttrCerts**. The **securityLabel** is associated with the MM that the CSP PDU is protecting as its **encapsulatedContent**, while the **kmAttrCerts** (containing the originator's attribute certificates) is associated with the originator's key management certificate. The association between an attribute certificate of the originator and his/her key management certificate is relatively strong. This issue shall not be discussed further in this paper except to say that the association is achieved via the binding created by the signature of the attribute certificate issuer on the attribute certificate.

The intended association between the **securityLabel** and the MM is the concern of this paper. It is evident from either [6] or [7] that this association is not direct and permanent in nature, but it depends on the association between the **messageSecurityData** (the field that includes the **securityLabel**) and the MM. Both the MM and the **messageSecurityData** must appear only in their encrypted form inside the CSP PDU during transportation. They share the same **msgKey** for the transformation into their encrypted form. When an intended recipient receives the CSP PDU for the first time, he/she can be assured of the association between the MM and the **messageSecurityData**. This follows because the intended recipient can recover the **msgKey** and restore both the MM and the **messageSecurityData** to their clear text form simultaneously. However, the recipient is not able to demonstrate this MM-**messageSecurityData** binding to a third party. We shall discuss this deficiency further in Section 2.5 (which addresses CSP PDU forwarding).

2.4 Functions Required at the Submission Port

Section 2.3 has explained briefly the construction of a CSP PDU for protecting a MM (P772 PDU), based on the security service indicators selected by the originator. The CSP process unit has the sole responsibility for this construction. After the construction is completed, the CSP PDU is passed to the message submission port, where the CSP PDU is included as the **content** of the **messageSubmission ARGUMENT**. The **messageSubmission ARGUMENT** may be referred as a P3 PDU where P3 is the protocol governing the interaction with a MTA. In addition, the **messageSubmission ARGUMENT** requires a **messageSubmissionEnvelope**.

As described in X.411 [12], the **messageSubmissionEnvelope** contains transportation-specific information objects such as originator-name, recipient-names, message-security-label, priority and latest-delivery-time. These information objects appear only in their clear text form and no special mechanism exists for ensuring their integrity. From a recipient's perspective, these information objects are less trustworthy than those belonging to the CSP **encapsulatedContent**, **signatureInformation**, **messageSecurityData**, and **recipientKeyToken**. They merely are used by the MTS to transport the **content** (CSP PDU), similar to the way that addressing information on a paper based envelope is used by the postal system. The completed **messageSubmission ARGUMENT** is then passed to the MTS via a local MTA (Figure 3).

In some cases, there may be a requirement to install a trusted message (CSP PDU) guard before the message submission port to interact with the MTA. It is expected that this message guard operates only at the CSP protocol level. It is not desirable for the message guard to access the MM heading field elements within the P772 PDU. Indeed, if the P772 PDU (inside the CSP **encapsulatedContent**) has been encrypted, the message guard will not be able to access any P772 PDU information objects (in the clear text form), which is the intent of the CSP "writer-to-reader" security, explained in Section 2.3. We shall not discuss the trusted message guard further in this paper. It suffices to say that this message guard ensures that only authorised CSP PDUs are allowed to exit from the originator's local area network. This is enabled by the use of **SIGNED Csp** (Figure 4) between the CSP process unit and the trusted message guard.

2.5 CSP PDU Forwarding

Forwarding generally involves CSP PDUs that previously have been created or received. We call each of these previously created or received CSP PDUs a forwarded CSP PDU. According to [7] or [6], CSP PDU forwarding requires that

1. a new MM is composed;
2. every forwarded CSP PDU is included in the new MM as a separate body part of type **Forwarded-CSP-Message-Body-Part**; and
3. a new CSP PDU is created by the CSP process unit to protect the new MM in its **encapsulatedContent**.

2.5.1 Alternative States of Forwarded CSP PDUs

There are two mutually exclusive states of a forwarded CSP PDU. The first is the encrypted state while the second is the clear text state. A CSP PDU could be received in either the encrypted or clear text state, prior to any subsequent CSP processing by the recipient.

2.5.1.1 Encrypted State

In the encrypted state (depicted in Figure 5), the information object inside the **encapsulatedContent** of the forwarded CSP PDU is encrypted. The **recipientSecurityData**

of the forwarded CSP PDU must be present to allow the information object inside the **encapsulatedContent** to be decrypted. The forwarded CSP PDU may or may not include any of the other fields (such as **originatorSecurityData**, **signatureBlock**, **contentDescription**, **mlExpansionHistory**, and **extensions**). If the **originatorSecurityData** is present and contains the **messageSecurityData**, then the **messageSecurityData** is necessarily encrypted. Similarly, if the **signatureBlock** is present and contains the **signatureValue**, then the **signatureValue** is necessarily encrypted (in accordance with [6] or [7]). Hence, the forwarded CSP PDU is considered to be unclassified by the definition of an encrypted material. However, it is not recommended that a forwarded CSP PDU in the encrypted state should be used for forwarding, unless the forwarded CSP PDU has remained “sealed” via a digital signature (or equivalent) since its original arrival and reception.

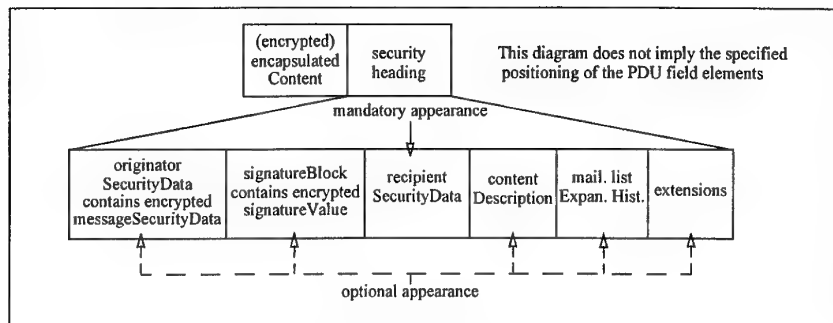


Figure 5 A Forwarded CSP PDU in the Encrypted State

The forwarded CSP PDU in the encrypted state actually presents opportunities for covert channel exploitation. The intermediate CSP process unit (which handles the forwarding) may or may not be able to decrypt, and subsequently understand, the encrypted information objects. Suppose that a trusted seal was not applied to the forwarded CSP PDU on its arrival and reception at the intermediate CSP process unit and that the intermediate CSP process unit has no other mechanism to verify that the integrity of the forwarded CSP PDU is preserved inside the local network. Then hostile software hidden inside the local network could substitute some classified information objects found in the local network for the encrypted information objects of the forwarded CSP PDU. These classified information objects could then exit unauthorised from the local network within the sabotaged forwarded CSP PDU. For this reason, we shall not focus on forwarded CSP PDUs in the encrypted state further. The other state (explained below) of forwarded CSP PDUs should be used for forwarding instead.

2.5.1.2 Clear Text State

In the clear text state (depicted in Figure 6), the information object inside the **encapsulatedContent** of the forwarded CSP PDU is in its clear text form. The forwarded CSP PDU may include the **signatureBlock** in addition to the **encapsulatedContent**. All the other fields (such as **originatorSecurityData**, **recipientSecurityData**, **contentDescription**, **mlExpansionHistory**, and **extensions**) are excluded explicitly. If the **signatureBlock** is included in the forwarded CSP PDU, then its **signatureValue** is necessarily in the clear text form. This **signatureValue** is associated with the clear text information object inside the **encapsulatedContent** and the **signatureInformation** inside the **signatureBlock**. The originator of the forwarded CSP PDU is the creator of this **signatureValue**.

The clear text state is the preferred form in which a CSP PDU is forwarded. The rest of this paper will focus mainly on the clear text state of a forwarded CSP PDU.

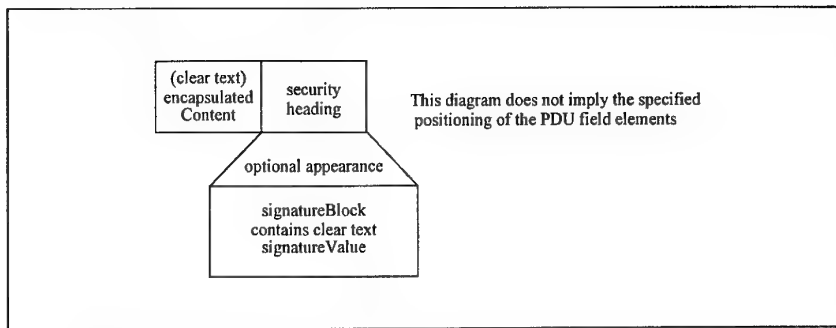


Figure 6 A Forwarded or Retained CSP PDU in the Clear Text State

The clear text state is also the preferred state in which a previously received or created CSP PDU should be retained. If a CSP PDU is already in the clear text state on its arrival and reception, then it is retained as is. Suppose that the CSP PDU is in the encrypted state on its arrival and reception. To retain it in the clear text state, processing is required to

- convert the information object (inside the CSP PDU **encapsulatedContent**) from its encrypted form into its clear text form;
- discard the **originatorSecurityData**, **recipientSecurityData**, **contentDescription**, **mlExpansionHistory**, and **extensions** fields from the CSP PDU;
- keep the **signatureBlock** if it already exists inside the CSP PDU; and
- convert the **signatureValue** in its encrypted form (namely the **encSigData**) into the **signatureValue** in its clear text form (namely the **sigValue**), where the **signatureValue** is a subfield of the **signatureBlock**.

This results in a retained CSP PDU in the clear text state. This retained CSP PDU is in the preferred state in which it is ready to be forwarded.

It should be noted that the retained CSP PDU (in the clear text state) does not include the **messageSecurityData** (and hence the **securityLabel**) which is associated with the information object inside its **encapsulatedContent**. As mentioned in Section 2.3.3, it is not possible to demonstrate the association between the **messageSecurityData** and the information object inside its **encapsulatedContent** to a third party. The **securityLabel**, therefore, cannot serve its purpose properly, even if it were to remain as part of the retained or forwarded CSP PDU (in the clear text state). Hence, the **originatorSecurityData** is deleted from the retained CSP PDU during its conversion to the clear text state.

A similar argument requires that the **recipientSecurityData**, **contentDescription**, and **mlExpansionHistory** also be deleted from the retained CSP PDU during its conversion into the clear text form. The case for the **extensions**, however, is slightly different. The decision for deletion depends on whether the **extensions** contains a subfield which has a permanent cryptography-based association with the clear text information object inside the **encapsulatedContent**. If the **extensions** does not contain such a subfield, then it is deleted from the retained CSP PDU during its conversion to the clear text state. On the other hand, if it contains such a subfield, then it is kept in the retained CSP PDU in the clear text state. Its treatment is similar to that of the **signatureBlock**, described above. Since the definition of the **extensions** is a national-specific matter, we shall not discuss its use further in this paper. For the purpose of this paper, we assume that the **extensions** does not contain a subfield which has a permanent cryptography-based association with the clear text information object inside the **encapsulatedContent**. Hence, the **extensions** is deleted from the retained CSP PDU during its conversion into the clear text state.

2.5.2 CSP PDU Forwarding Procedures

This subsection explains the CSP procedures, specific to the CSP PDU forwarding, where every forwarded CSP PDU contains a MM (namely a P772 PDU) in its **encapsulatedContent**. Suppose that Ada is a recipient who receives CSP PDU_1 containing a military message MM_1 from Bob. Assume that Ada wishes to forward MM_1 to a third party, say Cathy, as depicted in Figure 7.

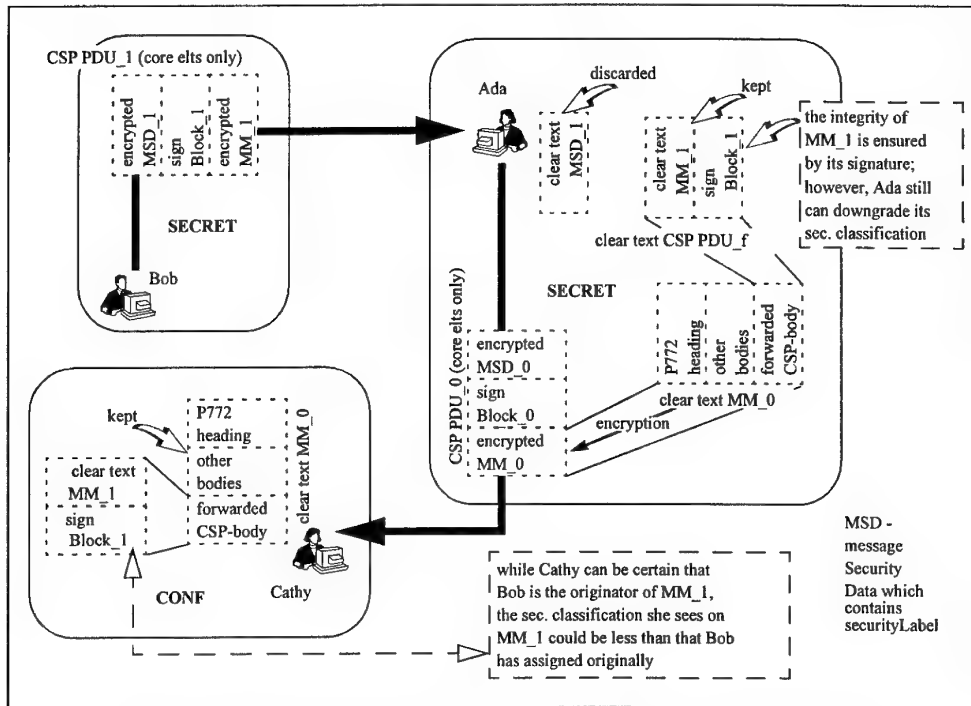


Figure 7 The Action Sequence Associated with MM Forwarding

As an authorised recipient of CSP PDU_1, Ada recovers

- the military message MM_1 in its clear text form;
- the associated **signatureBlock** signBlock_1 with its clear text **signatureValue**; and
- the **messageSecurityData** MSD_1 in its clear text form.

She may discard MSD_1 and the **securityLabel** (contained within) because the association between MSD_1 and MM_1 is not preserved beyond the clear text form recovery. Nevertheless, based on the definition of a retained or forwarded CSP PDU (Section 2.5.1.2), the combination of the clear text MM_1 and the signBlock_1 (with its clear text **signatureValue**) becomes (a retained as well as) a forwarded CSP PDU in the clear text state (Figure 6). Let us denote this CSP PDU in the clear text state by CSP PDU_f.

In the following discussion, we shall denote the new MM being composed, and the new CSP PDU being created (in a CSP PDU forwarding) as MM_0 and CSP PDU_0, respectively.

Suppose that Ada wishes to forward CSP PDU_f to Cathy. She composes MM_0 which includes CSP PDU_f as a separate body part. The type of this body part must be indicated explicitly as **Forwarded-CSP-Message-Body-Part** (Figure 7). Evidently, all information objects belonging to MM_0 (including CSP PDU_f which in turn includes MM_1 and signBlock_1) appear in their clear text form. They therefore require CSP protection during the transportation to Cathy. Specifically, MM_0 needs to be protected. This requires that Ada's CSP process unit creates CSP PDU_0 to encapsulate MM_0 (in its encrypted form) in its **encapsulatedContent** (Figure 7). Ada optionally may indicate the requirement of a

signature associated with MM_0. In that case, the **signatureBlock** signBlock_0 is included in CSP PDU_0. Specifically, signBlock_0 should contain

- the **signatureInformation** assigned by Ada for MM_0, and
- the **signatureValue** in its encrypted form (where the **signatureValue** is associated with MM_0 and the **signatureInformation**).

Optionally, Ada's CSP process unit may include the encrypted form of the **messageSecurityData** MSD_0 in CSP PDU_0 (Figure 7). However, MSD_0 need not share any relationship with MSD_1. There is no infosec mechanism which enforces the security requirement that the **securityLabel** contained in MSD_0 should be at least as restrictive as that contained in MSD_1.

Suppose that CSP PDU_0 has arrived at Cathy's local network. As an authorised recipient of CSP PDU_0, she recovers

- the military message MM_0 in its clear text form;
- the associated **signatureBlock** signBlock_0 with its clear text **signatureValue**; and
- the **messageSecurityData** MSD_0 in its clear text form.

Within MM_0 in its clear text form, she further recovers

- the military message MM_1 in its clear text form; and
- the associated **signatureBlock** signBlock_1 with its clear text **signatureValue**.

Using the information objects found in signBlock_0, Cathy is certain that MM_0 is originated from Ada. Similarly, using the information objects found in signBlock_1, she also is certain that MM_1 is originated from Bob. However, she cannot be certain that the securityLabel (in MSD_0) is exactly as Bob originally assigned it to MM-1.

2.5.3 *An Application of the Forwarding Mechanisms*

The above example has demonstrated the CSP PDU (and therefore MM) forwarding mechanisms, as currently allowed within the CSP specification [6] or the MSP specification [7]. It is intended that these forwarding mechanisms be used to achieve an electronic analogue of the composition of the paper based formal military correspondence, as described in Section 2.1.1. Recall from Figure 2 that a formal military correspondence may consist of

- one primary information item (such as (covering) minute, message, letter, directive, order, or operational plan); and
- one or more secondary information items (such as annexes or enclosures).

Each secondary information item may be written by (or originated from) some one else other than the writer or the originator of the primary information item. Even if all the primary and secondary information items are written by the same person, he/she may prefer to

- assign a unique security label to the primary information item and each secondary information item (as well as the collective whole of the military correspondence);
- assign an individual receipt instruction to the primary information item and each secondary information item (as well as the entire military correspondence); and
- individually sign the primary information item and each secondary information item (as well as the entire military correspondence).

This allows the writer a finer control over the first-tier distribution of his formal military correspondence consisting of the primary and secondary information items. Indeed, he/she also maintains the control over the m^{th} -tier follow-on-distribution of an individual information item belonging to his/her military correspondence. Suppose that a m^{th} -tier recipient receives only the primary information item or one of the secondary information items via the follow-on distribution. Because of the individual signature associated with the received information item, the recipient still can be certain about the origin and the

authenticity of the information item. This would not be possible if the writer/originator has signed only the collective whole of the military correspondence as a single compound object. This can be demonstrated in the general case as follows.

Each primary or secondary information item is treated as a single military message. Let us denote the primary information item by MM_p. Suppose that there are n secondary information items. Let us denote them by MM_s1, . . . , MM_sn. As military messages, some of them (namely, MM_p, MM_s1, . . . , MM_sn) may not be required to include any MM heading fields other than the **thisMM** field (for referencing purpose). Let CSP PDU_p, CSP PDU_s1, . . . , CSP PDU_sn be the CSP PDUs which encapsulate MM_p, MM_s1, . . . , MM_sn (in their clear text form), respectively, in their **encapsulatedContent** fields. Each of these CSP PDUs also includes its own **signatureBlock** field in addition to its **encapsulatedContent** field. This, therefore, allows the provision of an individually assigned **signatureInformation** and an individually computed (clear text) **signatureValue** to the primary item MM_p and each of the secondary information items MM_s1, . . . , MM_sn. CSP PDU_p, CSP PDU_s1, . . . , CSP PDU_sn are all retained CSP PDUs in the clear text state (as explained in Section 2.5.1.2). They are in the preferred state in which they are ready to be forwarded.

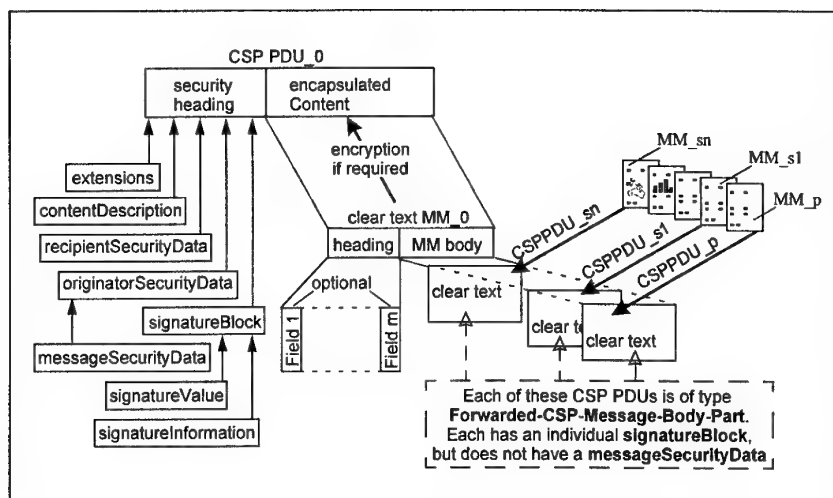


Figure 8 Electronic Analogue to a Paper Based Military Correspondence

A new military message (denoted by MM_0) is now composed by the writer to include CSP PDU_p, CSP PDU_s1, . . . , CSP PDU_sn as separate body parts (all of type **Forwarded-CSP-Message-Body-Part**) of MM_0. The writer also need to assign the MM heading field elements (such as **thisMM**, **originator**, **primaryRecipients**, **copyRecipients**, **related-MMs**, **precedence**, **messageInstruction**, **distributionCodes**, etc.) to MM_0 as required. Let CSP PDU_0 be the CSP PDU which encapsulates MM_0 (encrypted if required) in its **encapsulatedContent** field. This CSP PDU also includes

- its own **signatureBlock** field with the **signatureInformation** and **signatureValue** subfields (where the **signatureValue** must be encrypted also if MM_0 is encrypted);
- its optional **originatorSecurityData** field with the encrypted optional **messageSecurityData** subfield;
- its optional **recipientSecurityData** (in particular; it becomes mandatory if either MM_0 is encrypted or the **messageSecurityData** is required);
- its optional **contentDescription** field; and
- its optional **extensions** field

in addition to its **encapsulatedContent** field containing MM 0.

It is evident from Figure 8 that CSP PDU_0 almost becomes an electronic analogue to the paper based formal military correspondence (as depicted in Figure 2).

What essentially is missing from the above CSP PDU_0 composition is a CSP-based infosec mechanism which assigns and preserves an individual **securityLabel** for each of the forwarded military messages MM_p, MM_s1, . . . , MM_sn within CSP PDU_p, CSP PDU_s1, . . . , CSP PDU_sn respectively. Hence, we can only say that CSP PDU_0 almost becomes an electronic analogue to the paper based formal military correspondence. This is due to weaknesses (associated with the current CSP PDU forwarding mechanisms).

2.5.4 Some Security Weaknesses Associated with CSP PDU Forwarding

In view of the current CSP specification ([6] or [7]), there is no CSP-based infosec mechanism which enforces the following message forwarding-specific security policy.

The **securityLabel** assigned to any message which contains a forwarded message should be at least as restrictive as the **securityLabel** originally assigned to the forwarded message by the creator of the forwarded message.

This security policy requires that once a **securityLabel** has been assigned to a message by its creator, it should be preserved and associated permanently with the message (or its copies) until the message and all its copies are destroyed under authorisation. When the message is forwarded, a copy of the message effectively is created. It follows that the association between the **securityLabel** and the forwarded message should be maintained so that the **securityLabel** is not subject to modification by third parties.

The CSP submission access control is a determination made concerning the authorisation of the originator to send a message ([6] & [7]). Currently, the access control decision is based on the authorisation information of the originator, the recipients, the originator's end system, and the **securityLabel** that the originator has assigned to the message. Specifically, the CSP process unit checks that the **securityLabel** is within the range of the originator's end system, and that both the originator and recipient have the required authorisations. These checks are made for each recipient and a single failure results in the rejection of the message submission ([6] & [7]).

It is important to note that the current CSP submission access control does not require the CSP process unit check for any forwarded messages included (as a body part of type **Forwarded-CSP-Message-body-part**) within the message being submitted. Consequently, the CSP process unit is not required to ensure that the **securityLabel** of the message being composed actually dominates that of any of its forwarded messages. In fact, such a check would be pointless because the current structure of a CSP PDU does not provide a means for guaranteeing the integrity of the security label of a forwarded CSP PDU.

If the forwarded CSP PDU in question is in the encrypted state (Figure 5), then its **messageSecurityData** is encrypted. As the CSP process unit cannot decrypt the encrypted **messageSecurityData**, it cannot assess the **securityLabel** hidden cryptographically inside the **messageSecurityData**. In any case, we have recommended in Section 2.5.1.1 that a forwarded CSP PDU in the encrypted state should not be used for forwarding unless the forwarded CSP PDU has remained "sealed" via a digital signature (or equivalent) since its original arrival and reception.

In the other case, suppose that the forwarded CSP PDU in question is in the clear text state (Figure 6). It is evident from Figure 6 and Figure 7 that there is no **securityLabel** (which is associated with the forwarded CSP PDU **encapsulatedContent**) inside the forwarded CSP PDU. Without the **securityLabel**, it is impossible for the CSP access control mechanism to complete the security check on the **encapsulatedContent** of the forwarded CSP PDU.

As a result, the currently available CSP-based infosec mechanisms cannot enforce the above forwarding-specific security policy properly. The enforcement of this policy, therefore, must involve and depend on the integrity of the human user who initiates the message forwarding. Furthermore, whenever a previously received message needs to be forwarded to another user residing in an external network, a human reviewer is required to inspect the message before issuing an appropriate exit authority. This complication stems from the fact that previously received messages do not have an integrity-guaranteed **securityLabel** permanently associated with them during their life-cycle. This also may present further difficulties when the introduction of an automatic military message forwarding system is considered in the future.

A further weakness relates to control of dissemination of messages. Suppose that the original writer of a message does not wish the recipients to forward the message (which has been accompanied by the original writer's signature) to anyone outside the set of designated recipients. There is no means by which the current CSP submission access control could prohibit a recipient to act against this intention of the original writer.

To overcome the aforementioned weaknesses, Section 3 suggests the introduction of a permanent association between an integrity-guaranteed **securityLabel** and the clear text CSP PDU **encapsulatedContent** within the CSP protocol data unit structure (Figure 4).

This is a blank page.

3 Permanent Integrity-guaranteed Security Label

Recall from Figure 4 that the CSP security heading includes the **originatorSecurityData** and the **signatureBlock** fields. As their names imply,

- all information objects belonging to the **originatorSecurityData** field should be specific to the originator; and
- information objects belonging to the **signatureBlock** field should be specific to the CSP **encapsulatedContent** in its clear text form (namely the MM (P772 PDU) being composed).

The **messageSecurityData** and the **securityLabel** (contained within) clearly are information objects specific to the CSP **encapsulatedContent** in its clear text form. They, however, belong to the **originatorSecurityData** field. In addition, the **messageSecurityData** contains the **kmAttrCerts**, which is an information object specific to the originator's attribute certificates. However, the **kmAttrCerts** are not specific to the clear text CSP **encapsulatedContent**.

3.1 Proposed CSP PDU Structural Changes

This subsection proposes the following changes in the CSP protocol data unit sub-structure.

3.1.1 Changes in the OriginatorSecurityData Sub-Structure

The **kmAttrCerts** field is retained in the subfield-element-hierarchy of the **originatorSecurityData** field, but allocated at the uppermost level. In addition, the **messageSecurityData** field is removed from the **originatorSecurityData** field. We present the resultant of these changes in Figure 9.

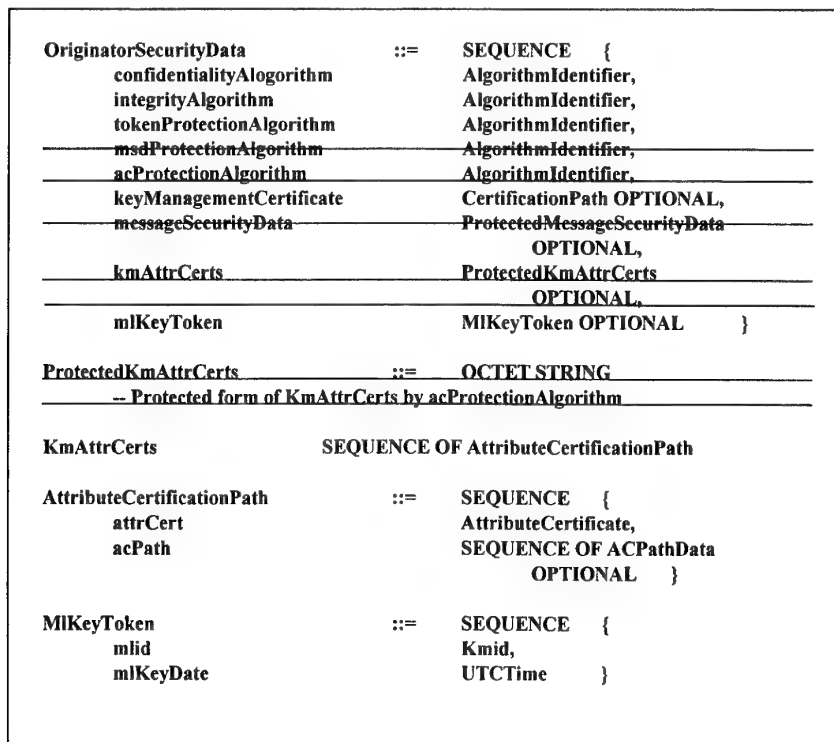


Figure 9 Proposed Changes to the **OriginatorSecurityData** Sub-Structure

Because the **messageSecurityData** field has been removed from the **originatorSecurityData**, the **kmAttrCerts** can no longer inherit the protection associated with the **messageSecurityData**. The **kmAttrCerts** field therefore requires its own protection algorithm, namely the **acProtectionAlgorithm**. It is expected that the message encryption key (**msgKey**) used for protecting the CSP PDU **encapsulatedContent**, the **messageSecurityData**, and the **signatureValue** will also be used for protecting the **kmAttrCerts**. However, a different initialisation vector may be used instead in this protection.

Actually, it is not clear if there truly exists the requirement for the **kmAttrCerts** protection. The only reason **kmAttrCerts** is currently protected is because it is located within the **messageSecurityData** which is always protected. If it transpires that the **kmAttrCerts** protection is not needed, then the **acProtectionAlgorithm** will simply be removed from the **OriginatorSecurityData** subfield-element-hierarchy, and the **kmAttrCerts** field will appear in its clear text form.

Finally, there is no change to the original structure or definition to the **keyManagementCertificate** or the **mlKeyToken**.

3.1.2 Changes in the SignatureBlock Sub-Structure

The modified **messageSecurityData** field (with the **kmAttrCerts** subfield removed from its subfield-element-hierarchy) is placed inside the **signatureBlock** subfield-element-hierarchy as shown in Figure 10.

The **messageSecurityData** is now allocated at the uppermost level of the **SignatureBlock** subfield-element-hierarchy. Its internal structure is modified because of

- the removal of the **kmAttrCerts** subfield; and
- the requirement for **messageSecurityData** to be in either its clear text state (**msdValue**) or its encrypted state (**encMsdData**).

This modified **MessageSecurityData** allows the choice of using either the **msdValue** or the **encMsdData**. The **encMsdData** is used when a new CSP PDU is being created for message transportation (Figure 4), while the **msdValue** is used for retaining a CSP PDU or for forwarding as a **Forwarded-CSP-Message-Body-Part**². The **encMsdData** is a sequence consisting of the **encMsdAlgorithm** and the **encMsdValue**. The **encMsdValue** is the **msdValue** in a protected (encrypted) form. Hence, the **encMsdValue** requires its own protection algorithm, namely, the **encMsdAlgorithm**. It is expected that the message encryption key (**msgKey**) used for protecting both the CSP PDU **encapsulatedContent** and **signatureValue** will also be used for transforming the **msdValue** into the **encMsdValue**. However, a different initialisation vector may be used for this protection.

In the current CSP PDU structure the **msdValue** is just a sequence which consists of only the **securityLabel**. If there is no requirement for including other information objects within the **msdValue** sequence³, then the **msdValue** simply can be made identical to the

-
2. This will be explained further in the following subsections.
 3. Referring to Figure 10, a viable alternative may be to remove the **sigAttrCerts** from the current allocation inside the **signatureBlock** and to relocate it inside the **msdValue** sequence. Recall that the existing **messageSecurityData** contains the **kmAttrCerts** according to the original specification of CSP [6] & [7]. An advantage in including the **sigAttrCerts** inside the **msdValue** is that both the signer's (or originator's) authority (as provided in the **sigAttrCerts**) and the **securityLabel** are bound to the message at the time the signature associated with the message is generated. However, further discussion of the inclusion of the **sigAttrCerts** inside the **msdValue** is outside the scope of this paper. For the purpose of this paper, we maintain the assumption that the **sigAttrCerts** is located inside the **signatureBlock** and not inside the **msdValue**.

securityLabel. In either case, there is no change to the original definition or structure of the **securityLabel**.

SignatureBlock	::=	SEQUENCE {
signatureAlgorithm		AlgorithmIdentifier,
signatureValue		SignatureValue,
controlInformation		ControlInformation,
messageSecurityData		MessageSecurityData OPTIONAL,
signatureCertificate		CertificationPath OPTIONAL,
sigAttrCerts		SEQUENCE OF
		AttributeCertificationPath
		OPTIONAL, }
SignatureValue	::=	CHOICE {
sigValue		SigValue,
encSigData		EncSigData }
ControlInformation	::=	CHOICE {
signatureInformation		SignatureInformation,
receiptInformation		ReceiptInformation }
SignatureInformation	::=	SEQUENCE {
encapsulatedContentType		CspContentType OPTIONAL,
signedContentIdentifier		OCTET STRING,
receiptRequests		ReceiptsIndicator,
receiptsTo		ORNameList OPTIONAL
		(SIZE (1..ub-receiptsTo)))
MessageSecurityData	::=	CHOICE {
msdValue		MsdValue,
encMsdData		EncMsdData }
EncMsdData	::=	SEQUENCE {
encMsdAlgorithm		AlgorithmIdentifier,
encMsdValue		ProtectedMsdValue }
ProtectedMsdValue	::=	OCTET STRING
-- Protected form of MsdValue		
MsdValue	::=	SEQUENCE {
label		SecurityLabel }
SecurityLabel	::=	SET {
security-policy-identifier		OBJECT IDENTIFIER OPTIONAL,
security-classification		SecurityClassification
		OPTIONAL,
privacy-mark		PrivacyMark OPTIONAL,
security-categories		SecurityCategories OPTIONAL }

Figure 10 Proposed Changes to the **SignatureBlock** Sub-Structure

There is also no change to the original structure and definition of the **signatureValue** which may still be either in the clear text state (**sigValue**) or in the encrypted state (**encsigData**). However, the definition of **sigValue** (in terms of the way that it is computed) will have to be changed because the hash computation will now have to take the **msdValue** into account. This is described in detail in Section 3.2.1.3.

There also is no change to the original structure and definition of the **controlInformation** which may be the **signatureInformation** (when the CSP PDU is created for a MM) or the **receiptInformation** (when the CSP PDU is created for a receipt). The **controlInformation** still is part of the **sigValue** computation.

Finally, there is no change to the original structure or definition of the **signatureCertificate** or the **sigAttrCerts**.

3.1.3 Changes to Forwarded and Retained CSP PDUs

Following the changes in the **OriginatorSecurityData** and the **SignatureBlock** described in the last two subsections, this subsection examines the implied changes to forwarded and retained CSP PDUs.

The changes to the basic structure of a forwarded CSP PDU in the encrypted state are depicted in Figure 11. They are explained as follows. The information object inside the **encapsulatedContent** of the forwarded CSP PDU is still encrypted. The **recipientSecurityData** of the forwarded CSP PDU is still mandatory, as the **encapsulatedContent** is encrypted. The forwarded CSP PDU may, or may not, include any of the other fields (such as **originatorSecurityData**, **signatureBlock**, **contentDescription**, **mlExpansionHistory**, and **extensions**). Recall that the optional inclusion of the **messageSecurityData** inside the **signatureBlock** (instead of inside the **originatorSecurityData**) is part of the proposed changes, described in Section 3.1.2. If the **signatureBlock** is present and it contains the **messageSecurityData**, then the **messageSecurityData** is necessarily encrypted. Similarly, if the **signatureBlock** is present and it contains the **signatureValue**, then the **signatureValue** is also encrypted. Hence, the forwarded CSP PDU in the encrypted state can still be considered to be unclassified for transmission purposes. It is still not recommended that a forwarded CSP PDU in the encrypted form should be used for forwarding, unless the forwarded CSP PDU has remained "sealed" via a digital signature (or equivalent) since its original arrival and reception.

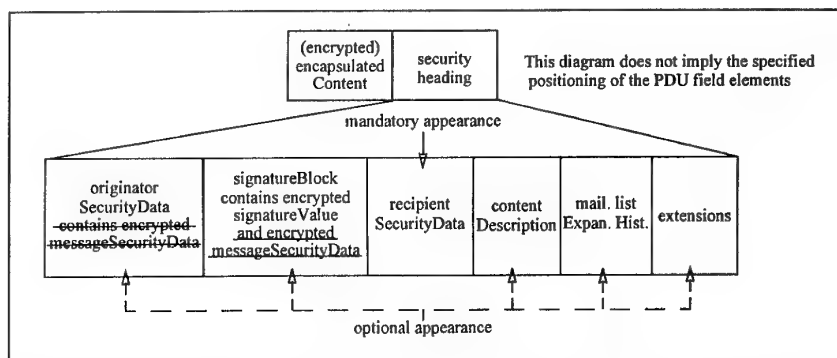


Figure 11 Implied Changes to a Forwarded CSP PDU in the Encrypted State

The implied changes to the basic structure of a forwarded CSP PDU in the clear text state are depicted in Figure 12. They are explained as follows. The information object inside the **encapsulatedContent** of the forwarded CSP PDU is still in its clear text form. The forwarded CSP PDU may still include the **signatureBlock** in addition to the **encapsulatedContent**. All the other fields (such as **originatorSecurityData**, **recipientSecurityData**, **extensions**, **contentDescription**, and **mlExpansionHistory**) still are excluded explicitly. If the **signatureBlock** exists and it contains the **messageSecurityData**, then the **messageSecurityData** is necessarily in the clear text form. If the **signatureBlock** is included in the forwarded CSP PDU, then its **signatureValue** is also in the clear text form. It will be seen in Section 3.2.1.3 that this **signatureValue** is associated with

- the clear text information object inside the **encapsulatedContent**; and
- the **signatureInformation** inside the **signatureBlock**; and
- the clear text **messageSecurityData** (if it exists) inside the **signatureBlock**.

The originator of the forwarded CSP PDU still is the creator of this **signatureValue**. The clear text state is still the preferred state in which a CSP PDU is forwarded.

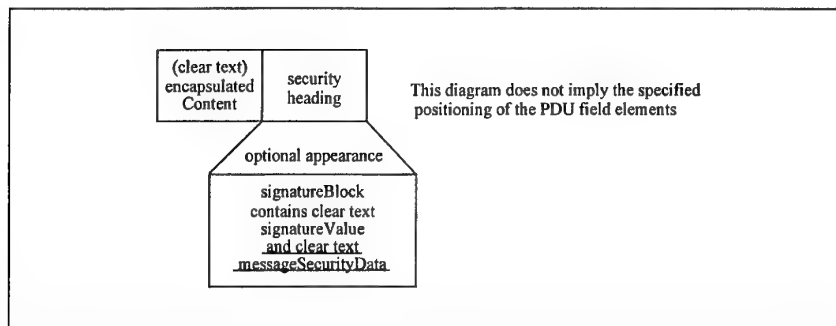


Figure 12 Implied Changes to Forwarded or Retained CSP PDU in the Clear Text State

The clear text state is still the preferred state in which a previously received or created CSP PDU should be retained. If a CSP PDU is already in the clear text state on its arrival and reception, then it is retained as it is. Suppose that the CSP PDU is in the encrypted state on its arrival and reception. To retain it in the clear text state, processing is required to

- convert the information object (inside the CSP PDU **encapsulatedContent**) from its encrypted form to its clear text form;
- discard the **originatorSecurityData**, **recipientSecurityData**, **contentDescription**, **mExpansionHistory**, and **extensions** fields from the CSP PDU;
- keep the **signatureBlock** if it already exists inside the CSP PDU;
- convert the **messageSecurityData** (if it already exists inside the **signatureBlock**) in its encrypted state (**encMsData**) to the **messageSecurityData** in its clear text state (**msdValue**);
- convert the **signatureValue** in its encrypted state (**encSigData**) to the **signatureValue** in its clear text form (**sigValue**), where the **signatureValue** belongs to the **signatureBlock**.

This results in a retained CSP PDU in the clear text state. This retained CSP PDU is in the preferred state in which it is ready to be forwarded.

3.2 Proposed Changes to the CSP PDU Handling Procedures

This subsection proposes the changes to the CSP protocol data unit handling procedures. These procedural changes complement the **originatorSecurityData** and **signatureBlock** structural changes explained in Section 3.1. All the **originatorSecurityData** and **signatureBlock** fields, and all the forwarded or retained CSP PDUs in the encrypted or clear text state mentioned in the following, conform to their respective structures and definitions as set out in Section 3.1. In particular, the **originatorSecurityData** conforms to Figure 9, the **signatureBlock** conforms to Figure 10, forwarded CSP PDUs in the encrypted form conform to Figure 11, and forwarded or retained CSP PDUs in the clear text form conform to Figure 12.

3.2.1 Changes To Message Submission

This subsection assumes that the originator is composing a new military message (denoted by **MM_0**) and his/her CSP process unit is creating a new CSP PDU (denoted by **CSP PDU_0**) to protect **MM_0**. It also is assumed that **MM_0** is a P772 PDU. There are two situations where the **messageSecurityData** field may be activated for **MM_0** and be included in **CSP PDU_0**.

1. The originator has indicated the use of **messageSecurityData** (in CSP PDU_0) for MM_0. He/she assigns the **securityLabel** (to MM_0) which is included in the **messageSecurityData**.
2. Suppose that MM_0 includes a forwarded CSP PDU (denoted by CSP PDU_f) in the clear text state in a body part of type **Forwarded-CSP-Message-Body-Part**. If CSP PDU_f contains its **messageSecurityData** in its **signatureBlock**, then CSP PDU_0 must also include **messageSecurityData** in its **signatureBlock**. This has to be the case regardless of whether the originator has chosen, or indicated to use, the **messageSecurityData** in CSP PDU_0 or not. Furthermore, the **messageSecurityData** (for MM_0) contained in CSP PDU_0 must be as restrictive as the available **messageSecurityData** contained in the forwarded CSP PDU (in the clear text state) inside every body part (of type **Forwarded-CSP-Message-Body-Part**) of MM_0. This ensures the enforcement of the message forwarding-specific security policy as discussed in Section 2.5.4.

3.2.1.1 *Consequences of MessageSecurityData Activation*

In either of the above two situations, the **signatureBlock** must be included in CSP PDU_0 because the **messageSecurityData** for MM_0 is included (Figure 10). Since the **signatureValue** is a mandatory subfield of the **signatureBlock**, it follows that activation of **messageSecurityData** for MM_0 implies signature generation for MM_0 regardless of whether the originator has requested the signature generation or not. In other words, the originator is prohibited from assigning a security label to MM_0 and then leaving MM_0 unsigned.

This restriction appears reasonable. If the originator wishes to assign a security label to MM_0, then he/she must demonstrate his/her authority to do so. This authority demonstration is exercised through the verification of **signatureValue** based on the associated **signatureCertificate** and **sigAttrCerts** contained in the **signatureBlock**.

Recall that CSP PDU_f is a forwarded CSP PDU in the clear text state. Suppose that CSP PDU_f contains its own **messageSecurityData** (Figure 12). Then CSP PDU_0 must also contain its own **messageSecurityData** (referring to Situation 2 above). For the same reasons as explained above, if the originator wishes to include CSP PDU_f in MM_0, then a signature for MM_0 has to be generated (automatically by the CSP process unit). In other words, the originator is prohibited from including a forwarded CSP PDU (which has its own security label) in MM_0 and then leave MM_0 unsigned.

Hence, the CSP process unit automatically assigns a **messageSecurityData** to MM_0, if MM_0 includes at least one forwarded CSP PDU (which has its own **messageSecurityData**). The **messageSecurityData** assigned to MM_0 (by the CSP process unit) must be at least as restrictive⁴ as those of the forwarded CSP PDUs included in MM_0, or that assigned to MM_0 by the originator⁵.

In the case of an unclassified message, the originator may still sign MM_0, where its forwarded CSP PDUs do not have their own **messageSecurityData** subfields and a

-
4. We define the meaning of the term “restrictive” associated with the **messageSecurityData** as follows. If one of the forwarded CSP PDUs contains a security classification, then the security classification of MM_0 is the highest classification among the forwarded CSP PDUs. In addition, if one of the forwarded CSP PDUs contains a security caveat, then the caveat of MM_0 is the combination of all the caveats of the all forwarded CSP PDUs.
 5. This ensures that the originator has assigned the **messageSecurityData** consistently. If the originator has not assigned the **messageSecurityData** consistently, then the CSP process unit (as a trusted element) should derive an appropriate **messageSecurityData** instead.

messageSecurityData subfield has not been assigned to MM_0. In this case, MM_0 is considered to be unclassified. In other words, there is no restriction on an originator signing (or not signing) any unclassified message.

3.2.1.2 *CSP Submission Access Control Determination*

Recall that CSP PDU_0 is the new CSP PDU created by the CSP process unit to protect MM_0. Based on the presence of **messageSecurityData** inside CSP PDU_0, or the absence of **messageSecurityData** from CSP PDU_0, the CSP submission access control determination is made. The CSP submission access control checks the authorisation information of the originator, the recipients, the originator's end system, and either

- the explicit **securityLabel** (that has been assigned to MM_0 via the **messageSecurityData**); or
- the implicit unclassified security label implied by the absence of **messageSecurityData** from MM_0, subject to the local security policy.

As explained in Section 2.5.4 above, a failure of this check for any recipient results in the rejection of the message submission.

3.2.1.3 *Implied Changes to Signature Generation*

Recall that MM_0 is the clear text form of the **encapsulatedContent** of CSP PDU_0. This subsection describes the generation of the signature (which is the **sigValue**) associated with MM_0. The following paragraphs explain the completion of the **SignatureBlock** of the CSP PDU_0 (and the assignment of its information objects contained within the CSP PDU_0).

As the first step, a complete hash over MM_0 is generated. This hash is the **msgHash**. If confidentiality has been invoked or the **messageSecurityData** is present inside CSP PDU_0, then the **msgHash** is placed inside the **RecipientKeyToken**. Subsequently, a second hash value (denoted by **msg&SignInfoHash**) is calculated over the concatenation of the **msgHash** and the **signatureInformation** found inside the **controlInformation** of the **SignatureBlock** (Figure 10).

In the second step, there are two alternative cases in which the signature generation may progress. The choice between the two cases depends on whether the **messageSecurityData** is present or absent from CSP PDU_0.

Case 1.

If the **messageSecurityData** is absent from CSP PDU_0, then the **msg&SignInfoHash** is used as the input to the **signatureAlgorithm**, which the CSP process unit uses to calculate the **sigValue**. If confidentiality has been invoked, the **sigValue** is transformed into the **encSigValue**, via the encryption **encSigAlgorithm** using the **msgKey**. The sequence consisting of the **encSigAlgorithm** and **encSigValue** then forms the **encSigData**. The **encSigData** consequently is used as the **signatureValue**. The hash calculation and signature generation for this case are depicted in Figure 13a. If, however, confidentiality is not invoked, it suffices that the **sigValue** is chosen to be the **signatureValue**. In this case, the hash calculation and signature generation are depicted in Figure 13b.

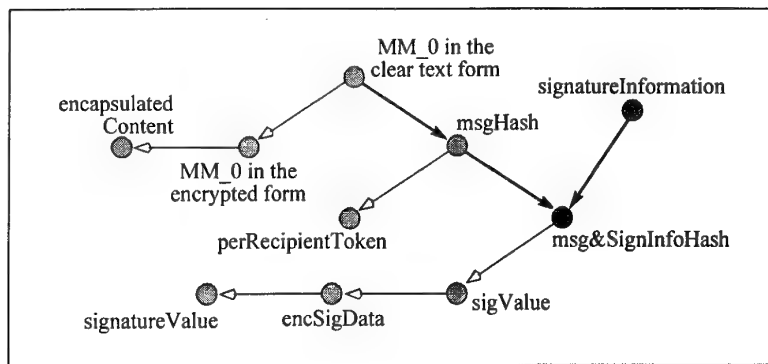
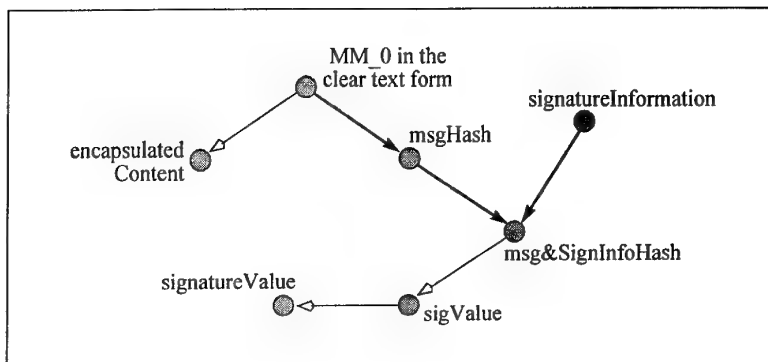
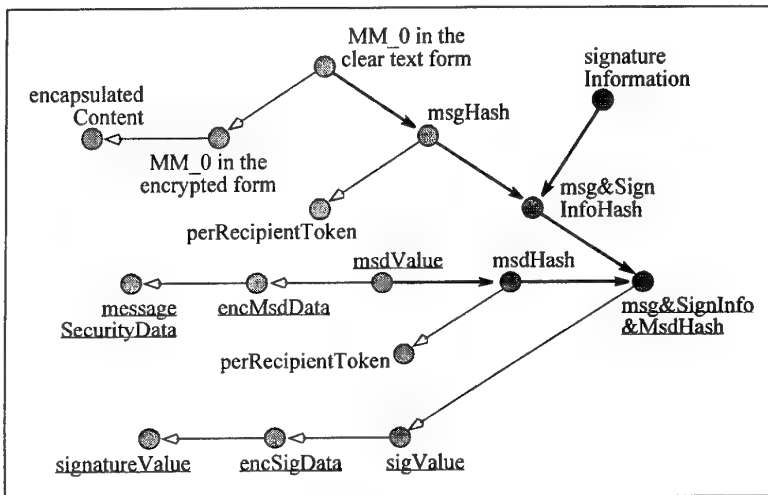
Figure 13a (Confidentiality Invoked & **messageSecurityData** Absent)Figure 13b (Confidentiality not Invoked & **messageSecurityData** Absent)Figure 13c (**messageSecurityData** Present)

Figure 13 Implied Changes to Hash Calculation & Signature Generation

Case 2.

If the **messageSecurityData** is present inside CSP PDU_0, then a hash is calculated over the **msdValue**. This hash is the **msdHash**, which is placed inside the **Recipi-**

entKeyToken. A final hash (denoted by **msg&SignInfo&MsdHash**) is calculated over the concatenation of the **msg&SignInfoHash** and the **msdHash**. The **msg&SignInfo&MsdHash** then is used as the input to the **signatureAlgorithm**, which the CSP process unit uses to calculate the **sigValue**.

Information objects belonging to the **messageSecurityData** cannot appear in their clear text form during transportation. Consequently, only the **encMsdData** can be chosen as the **messageSecurityData**. The **encMsdValue** is the result of the encryption transformation **encMsdAlgorithm** which is applied to the clear text **msdValue** with the **msgKey**. The sequence consisting of the **encMsdAlgorithm** and **encMsdValue** then forms the **encMsdData**. The presence of the **messageSecurityData** in CSP PDU_0 also implies that it is MM_0 in its encrypted (and not clear text) form that is placed in the **encapsulatedContent** of CSP PDU_0. It therefore is similar to the case where confidentiality has been invoked. Hence, the **sigValue** also needs to be transformed into the **encSigValue**, via the **encSigAlgorithm** with the **msgKey**. The **encSigData** then has to be used as the **signatureValue**. The hash calculation and signature generation for this case are depicted in Figure 13c.

As a final remark of this subsection, we note that Figure 13c in fact depicts the inter-relationships between various signature-specific information objects of the CSP PDU CSP PDU_0 being created by the originator's CSP process unit. It is evident that, when it arrives, after forwarding, at a recipient, CSP PDU_0 has exactly the structure of a forwarded CSP PDU in the encrypted state (Figure 11). As described in Section 3.1.3, if CSP PDU_0 is retained by an intended (or authorised) recipient, its conversion into the clear text state is required. Although the conversion itself has been explained in Section 3.1.3, it is worthwhile to outline the inter-relationships between various information objects of the CSP PDU_0 structure within the retained or forwarded CSP PDU in the clear text state. We depict these inter-relationships in Figure 14.

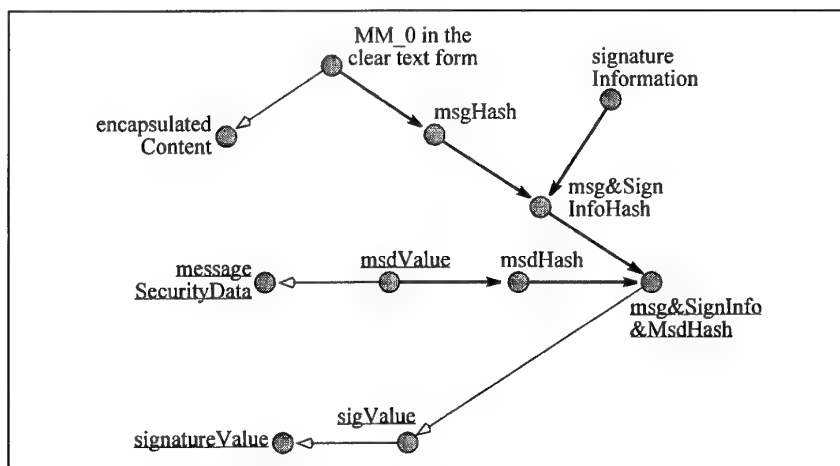


Figure 14 Relationships between objects of a Forwarded CSP PDU in the Clear Form

3.2.2 Changes to Message Reception and Forwarding

In this subsection, we revisit the example (described in Section 2.5.2) where Ada receives a message from Bob and she then forwards the message to Cathy.

Recall that Ada is a recipient who receives the CSP PDU CSP_PDU_1 in the encrypted state (where CSP_PDU_1 contains the military message MM_1 in its **encapsulatedContent**) from Bob. She then wishes to forward MM_1 to Cathy. As a result of the comparison with the action sequence described in Figure 7 (based on the original CSP specification before the proposed changes explained in Sections 3.1 & 3.2), the changes to the action sequence associated with the forwarding are depicted in Figure 15.

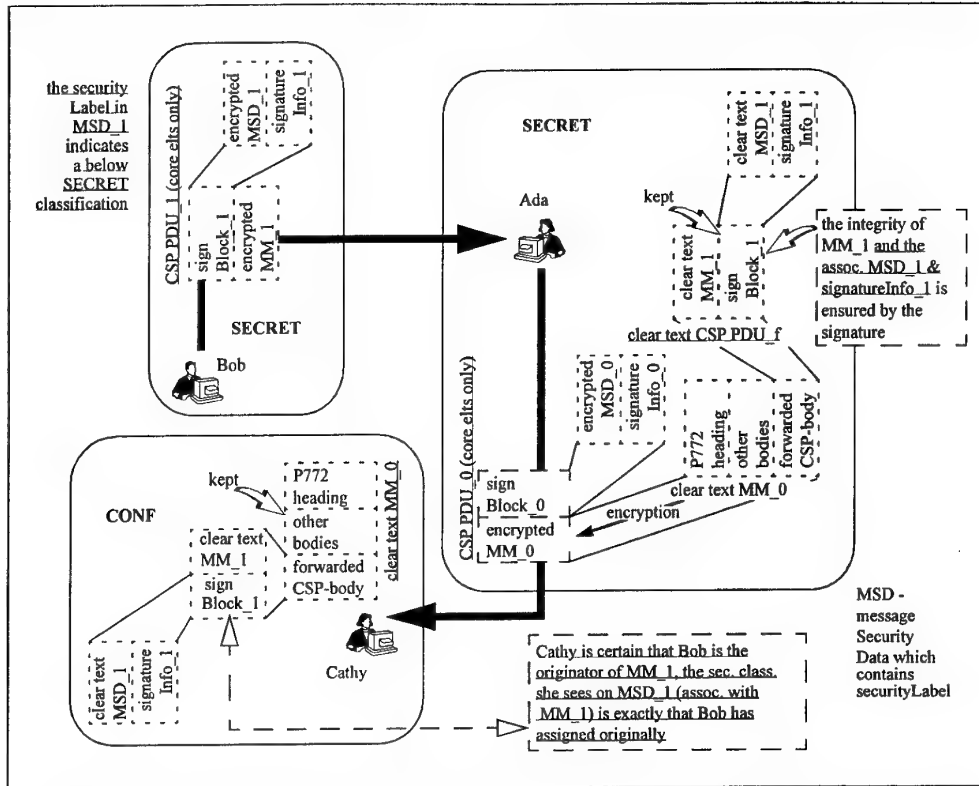


Figure 15 Changes to the Action Sequence Associated with MM Forwarding

As an authorised recipient of CSP_PDU_1 , Ada still recovers

- the military message MM_1 in its clear text form; and
- the associated **signatureBlock** $signBlock_1$ (which contains its clear text **signatureValue**, the **signatureInformation** $signInformation_1$ (always in its clear text), and the **messageSecurityData** MSD_1 in its clear text form).

In this case, not only is $signInformation_1$ bound to MM_1 , but also MSD_1 is bound to MM_1 via the **signatureValue**. Both $signInformation_1$ and MSD_1 are part of $signBlock_1$. Ada no longer can discard MSD_1 and the **securityLabel** (contained within) because the association between MSD_1 and MM_1 is preserved permanently, even after their clear text form recovery. Based on the changed definition of a retained or forwarded CSP PDU (Section 3.1.3), the combination of the clear text MM_1 and the $signBlock_1$ (with its clear text **signatureValue**, its clear text MSD_1 , and its $signInformation_1$) becomes (a retained as well as) a forwarded CSP PDU in the clear text state (Figure 12). We denote this CSP PDU in the clear text state by CSP_PDU_f .

When Ada wishes to forward CSP_PDU_f to Cathy, she compose a new military message MM_0 which includes CSP_PDU_f as a separate body part. The body part type must still be indicated explicitly as **Forwarded-CSP-Message-Body-Part** (Figure 15). All information objects belonging to MM_0 (including CSP_PDU_f , which in turn includes MM_1 and $signBlock_1$) are still in their clear text form. Hence, MM_0 needs to be protected. It

therefore requires that Ada's CSP process unit creates CSP PDU_0 to encapsulate MM_0 (in the encrypted form) in its **encapsulatedContent** (Figure 15). It follows from Section 3.2.1.1 that a signature associated with MM_0 now is mandatory because the forwarded CSP PDU (namely CSP PDU_f) contains a **messageSecurityData** field (namely MSD_1). Therefore, the **signatureBlock** signBlock_0 is included in CSP PDU_0. Specifically (based on Figure 10 & Figure 13c), signBlock_0 should contain

- the **signatureInformation** signInformation_0 assigned by Ada for MM_0;
- the **messageSecurityData** MSD_0 in its encrypted state (where MSD_0 is assigned by Ada or is generated automatically by Ada's CSP process unit); and
- the **signatureValue** in its encrypted state (where the **signatureValue** is associated with MM_0, signInformation_0, and MSD_0 as described in Figure 13c).

In determining the CSP submission access control, Ada's CSP process unit ensures that the **securityLabel** contained in MSD_0 is at least as restrictive as that contained in MSD_1. As a result, this becomes the CSP-based infosec mechanism which can enforce the forwarding-specific security policy stated Section 2.5.4 (namely that the **securityLabel** contained in MSD_0 should be at least as restrictive as that contained in MSD_1).

Suppose that CSP PDU_0 has arrived at Cathy's local network. As an authorised recipient of CSP PDU_0, she recovers

- the military message MM_0 in its clear text form; and
- the associated **signatureBlock** signBlock_0 (containing its clear text **signatureValue**, signInformation_0 (always in its clear text), and MSD_0 in its clear text form),

where both signInformation_0 and MSD_0 are bound to MM_0 via the **signatureValue**.

Within MM_0 in its clear text form, she further recovers

- the military message MM_1 in its clear text form; and
- the associated **signatureBlock** signBlock_1 (containing its clear text **signatureValue**, signInformation_1 (always in its clear text), and MSD_1 in its clear text form),

where both signInformation_1 and MSD_1 are bound to MM_1 via the **signatureValue**.

Using the information objects found in signBlock_0, Cathy can be certain that MM_0 is originated from Ada. Similarly, using the information objects found in signBlock_1, she can also be certain that MM_1 is originated from Bob. In addition, she also can be certain that the **securityLabel** belonging to MSD_1 is exactly as Bob originally assigned it to MM_1.

3.3 *A Closer Electronic Analogue to Paper Based Formal Military Correspondence Composition*

Recall from Section 2.1.1 the requirement for an electronic analogue to the composition of the paper based formal military correspondence. An attempt to realise the analogue based on the available CSP-based mechanisms and functionality (before the proposed changes explained in Sections 3.1 & 3.2) is only partially successful as shown in Sections 2.5.3 & 2.5.4. Based on the proposed changes to

- the CSP PDU sub-structures and associated definitions; and
- the CSP PDU handling procedures

described in Section 3.1 and Section 3.2 respectively, this subsection attempts to improve on the realisation of the electronic analogue to the composition of the paper based formal military correspondence.

Recall from Section 2.5.3 that MM_p is the primary information item and MM_{s1}, \dots, MM_{sn} are n secondary information items of a formal military correspondence. Let CSP PDU_p, CSP PDU_{s1}, ..., CSP PDU_{sn} be the forwarded CSP PDUs (in the clear text state) which encapsulate $MM_p, MM_{s1}, \dots, MM_{sn}$ (in their clear text form), respectively, in their **encapsulatedContent** fields. Each of these CSP PDUs includes its own **signatureBlock** in addition to its **encapsulatedContent**. According to the proposed changes as described in Section 3.1.3 and depicted in Figure 12, each **signatureBlock** contains its own (clear text) **messageSecurityData** as well as its own **signatureInformation** and **signatureValue**. This, therefore, allows the provision of an individually assigned **signatureInformation** and an individually assigned **messageSecurityData** (with a unique **securityLabel** contained within) to the primary item MM_p and to each of the secondary information items MM_{s1}, \dots, MM_{sn} . On the per-CSP PDU basis, the (clear text) **signatureValue** of each aforementioned CSP PDU is cryptographically associated with the clear text **encapsulatedContent**, the **signatureInformation**, and the clear text **messageSecurityData** of the CSP PDU as shown in Figure 14.

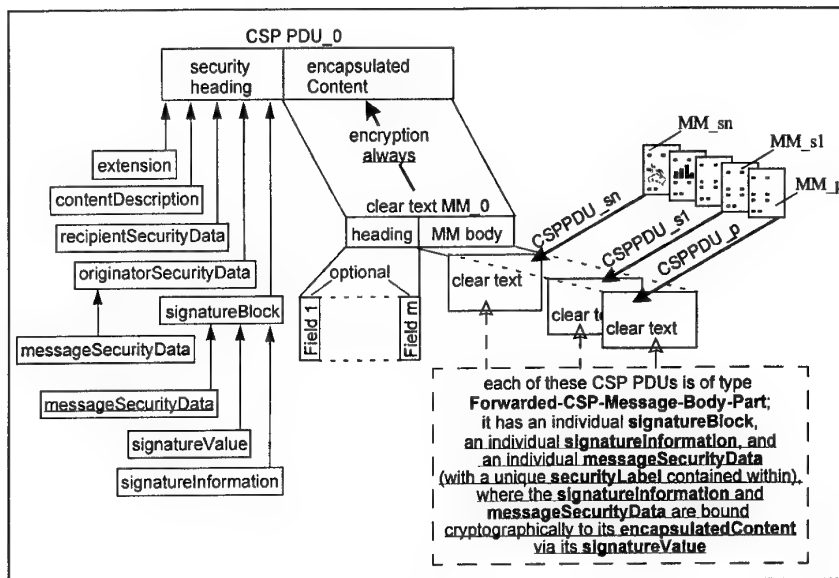


Figure 16 Improved Electronic Analogue to a Paper Based Military Correspondence

According to Section 2.5.3, the military message MM_0 is composed by the writer to include CSP PDU_p, CSP PDU_{s1}, ..., CSP PDU_{sn} as separate body parts (all of type **Forwarded-CSP-Message-Body-Part**). The writer assigns the MM heading field elements to MM_0 as required. Let CSP PDU₀ be the CSP PDU which necessarily encapsulates MM_0 in its encrypted form in its **encapsulatedContent** field. Recalling from Section 3.2.1.3 (and specifically its Case 2), CSP PDU₀ includes

- its (mandatory) **signatureBlock** field with its **signatureInformation**, **messageSecurityData**, and **signatureValue** subfields (where the **signatureValue** and **messageSecurityData** must be encrypted);
- its (mandatory) **recipientSecurityData** (because the **messageSecurityData** is encrypted);
- its (mandatory) **originatorSecurityData** field (because the **recipientSecurityData** is mandatory);
- its optional **contentDescription** field; and
- its optional **extensions** field

in addition to its **encapsulatedContent** field. The inter-relationships between the various signature-specific information objects of CSP PDU_0 are exactly those depicted in Figure 13c.

Because of the binding of a security label to every forwarded CSP body part, it is evident that CSP PDU_0 is now closer to become an electronic analogue to the paper based formal military correspondence (as depicted in Figure 2) than it was before applying the proposed changes to the CSP PDU sub-structures, sub-definitions, and handling procedures. We present this evidence graphically in Figure 16.

It should be emphasized that CSP PDU_0 has not actually become the analogue yet. There remains the need for a similar function to copy numbers appearing on the paper based accountable formal military correspondences. We shall leave the investigation associated with the copy number functionality as the subject of a future research paper.

3.4 Consideration of Other Approaches

Before proposing the changes to the **originatorSecurityData** and the **signatureBlock** (namely the relocation of the **messageSecurityData** from the **originatorSecurityData** to the **signatureBlock**) described in Sections 3.1 & 3.2, other approaches for accomplishing a similar electronic analogue to the paper based formal military correspondence (as depicted in Figure 2) were considered. One approach is the inclusion of the **securityLabel** (as a field element) inside the MM (P772 PDU) heading structure. However, this approach is considered to be less effective for the following reasons.

1. From the military perspective, the security label binding generally should have a wider application to many other (command and control) information items than to just messages (MM P772 PDU or IPM P22 PDU). The inclusion of the **securityLabel** inside the MM (or IPM) heading is just a particular solution for introducing the **securityLabel** into a MM P772 PDU or IPM P22 PDU. There are other information items such as voice, graphics, maps, demographic data, complex documents (including operational plans, intelligence reports or summaries, soldier's handbooks, weapons recognition guides, and press releases), and (more significantly) general military data base entry items. These information items generally are produced by commercial off the shelf products and are not necessary subject to international or military standardisation. It would be difficult to persuade the commercial vendors to (optionally or otherwise) embed the military-specific **securityLabel** based system into information items produced by their products.

Even if this is possible, the (necessarily trusted) CSP process unit still requires access to the embedded **securityLabel** in order to make the CSP submission (or delivery) access control determination. A trusted unit such as the CSP process unit is necessary simple in terms of its functionality because of the relatively higher assurance evaluation requirement. Accessing the **securityLabel** embedded inside an information item requires the CSP process unit to understand the data structure of the information item. This clearly increases the complexity of the CSP process unit and makes it relatively more difficult to be evaluated to its desired assurance level.

By comparison, the approach proposed in this paper (as explained in Sections 3.1, 3.2 & 3.3) simply makes use of the **securityLabel** and **signatureInformation** already available inside the CSP PDU structure. It only requires the encapsulation of an information item inside its **encapsulatedContent** and then have the "concatenation" of the information item, its associated **securityLabel** and **signatureInformation** signed to produce a **signatureValue** (as shown in Figure 13c & Figure 14). It does not matter what data structure the information item may have. This, therefore, implies a future proof approach in which the information security services (including the **securityLabel** based access control) of CSP (in future) may support other command and control appli-

cations as well as military messaging for which CSP was originally designed.

2. It is possible to consider an information item such as a MM P772 PDU (with just a single body part) as we have done in Figure 8 of Section 2.5.3 and in Figure 16 of Section 3.3. In these cases, the actual information item is the body part of the MM P772 PDU. Auxiliary information objects (such as **thisMM** for referencing purpose) associated with the information object are indicated in the heading of the MM P772 PDU. However, the MM P772 PDU still requires the CSP-based security protection as we have shown in Figure 8 of Section 2.5.3 and in Figure 16 of Section 3.3. Hence the **securityLabel** utilisation at the CSP level allows for uniform treatment of CSP (submission or delivery) access control to an information item, regardless of whether
 - the information item is encapsulated directly inside the **encapsulatedContent** of a CSP PDU; or
 - the information item initially is included in a MM P772 PDU and that MM P772 PDU subsequently encapsulated inside the **encapsulatedContent** of a CSP PDU.

Furthermore, by focusing on the trusted **securityLabel** provision at the CSP level, we maintain the operation of the information security services (including the **securityLabel** based access control) at the CSP protocol level rather than spreading their functionality to other protocols such as P772.

3. The attempt to include the **securityLabel** inside the heading of an IPM P22 PDU or a MM P772 PDU (through the international or military standardisation) has neither been successful in the development of X.400 [9], nor the development of STANAG 4406 [11]. The closest agreement (that has been reached by the standards bodies) is only the inclusion of the **sensitivity** (which is a much weaker mechanism than the **securityLabel** is) inside the heading of an IPM P22 PDU and a MM P772 PDU. In fact, according to [12] and [11], the **securityLabel** is provided as the **messageSecurityLabel** on the P3 **messageSubmissionEnvelope**. We have explained in Section 2.4 that the information objects (including the **messageSecurityLabel**) of the P3 **messageSubmissionEnvelope** are not trusted sufficiently for our purpose. This stems from the ITU and NATO security architectural approaches to X.400 based message transfer systems which differ in a fundamental way to the US, AUS and (generally) CCEB approaches. The ITU or NATO approach requires the implementation of trusted message transfer systems in which the **messageSecurityLabel** and the other information objects of the P3 **messageSubmissionEnvelope** obviously become effective. However, as it has been assumed throughout this paper, the US, AUS and (generally) CCEB approaches do not presuppose a trusted message transfer system. This is the main reason why CSP has been designed specifically to address the writer-to-reader message security in the environment of interconnected untrusted message transfer systems. For the reasons explained above, it is difficult to envisage the **securityLabel** inclusion inside the heading of an IPM P22 PDU or a MM P772 PDU gaining CCEB approval at any time in the near future.

4 Conclusion

Common Security Protocol (CSP) [6] or its technical equivalent Message Security Protocol (MSP) [7] has been designed specifically to enable the writer-to-reader security for military messaging, particularly that based on [8] and [9]. The crucial function of CSP is its encapsulation of a message within its PDU structure (namely inside its **encapsulatedContent**). Information security (infosec) services such as:

- message confidentiality;
- non-repudiation with proof of message origin authentication;
- non-repudiation with proof of message delivery; and
- message submission or delivery access control

are provided to the encapsulated message through the appropriate utilisation of the basic CSP infosec protocol mechanisms such as:

- exchange of message encryption keys for authorised access of encrypted messages (i.e. via its **recipientSecurityData**);
- generation of an originator's signature associated with the encapsulated message (i.e. via the **signatureInformation** and **signatureValue** belonging to its **signatureBlock**);
- generation of a recipient's signature associated with a message receipt (i.e. via the **receiptInformation** and **signatureValue** belonging to its **signatureBlock**); and
- security labelling of the encapsulated message (i.e. via the **securityLabel** belonging to the **messageSecurityData** inside its **originatorSecurityData**).

This paper has explored the application of CSP based infosec to a more general class of command and control information item than the basic military message. This class is the formal military correspondence covering commander/minister/secretary minutes or letters, command and control directives, and military operational orders or plans. A formal military correspondence typically consists of a primary part and a number of secondary parts as annexes or enclosures. Each of these annexes or enclosures may be written by, or have originated from, someone other than the correspondence originator. In addition, each annex or enclosure could be assigned its own unique security classification (or, more generally, security label) which may be different from that of the overall military correspondence. In some cases, a military correspondence also needs to be treated as an accountable document (which imposes further restraints in terms of the security requirements).

Through the examination of the CSP-based message forwarding function and its associated infosec protocol mechanisms, this paper has demonstrated that the CSP-based forwarding function (as currently defined in [6] or [7]) is not sufficient to provide an effective electronic analogue to the paper-based formal military correspondence composition.

Specific changes to the CSP PDU structure, and corresponding changes to the CSP handling procedures have therefore been proposed. The structural changes involve a simple relocation of the **messageSecurityData** from the **originatorSecurityData** to the **signatureBlock**. The procedural changes required are as follows:

- the activation of the **messageSecurityData** triggers the generation of a signature;
- the signature generation takes the **msgHash**, **signatureInformation** and the **msdHash** into account;
- the presence or activation of a **messageSecurityData** field in a Forwarded-CSP-Body-Part automatically activates the **messageSecurityData** of the super-ordinate CSP PDU; and
- the **securityLabel** within the super-ordinate CSP PDU must be as restrictive as the **securityLabel** contained within a subordinate forwarded CSP PDU.

This paper has demonstrated that, with the above proposed enhancements to CSP, a closer approximation to formal military correspondence composition can be achieved by an electronic analogue.

5 *References*

- [1] Australian Defence Force Publications, Operations Series, Operations (ADFP 6). Mar. 1996.
- [2] Australian Defence Force Publications, Operations Series, Joint Planning (ADFP 9). Apr. 1994.
- [3] Australian Defence Force Publications, Operations Series, Intelligence (ADFP 19). Apr. 1995.
- [4] Australian Defence Force Publications, Operations Series, Surveillance and Reconnaissance (ADFP 29). May 1995.
- [5] The US President Executive Order 12958 'Classified National Security Information'. The White House. 17 Apr. 1995.
- [6] Allied Communications Publication 120, Common Security Protocol, ACP120, Draft, 18 Mar. 1996.
- [7] SDNS Secure Data Network System Message Security Protocol (MSP) Specification, Revision 4.0. SDN.701. 18 Mar. 1996.
- [8] Allied Communications Publication 123, Common Messaging Strategy and Procedures, ACP 123, Military Communications Electronics Board. Nov. 1994.
- [9] CCITT Recommendation X.400 (1988) | ISO/IEC 100211 'Information Processing Systems - Text Communication - Message Oriented Text Interchange System'. 1988.
- [10] CCITT Recommendation X.420 (1988) | ISO/IEC 100217 'Information Processing Systems - Text Communication - Message Oriented Text Interchange System - Part 7: Interpersonal Messaging System'. 1988.
- [11] NATO STANAG 4406, Military Message Handling System, Version 2, Defense Information Systems Agency, September 1994.
- [12] CCITT Recommendation X.411 (1988) | ISO/IEC 100214 'Information Processing Systems - Text Communication - Message Oriented Text Interchange System - Part 4: Message Transfer System: Abstract Service Definition and Procedures'. 1988.

This is a blank page.

**Common Security Protocol Security Labelling
and its Applications**

M.K.F. Lai, J. Burgess, K. Forrest, H. Daniel & N.F. Parker

(DSTO-RR-0086)

DISTRIBUTION LIST

Number of Copies

AUSTRALIA

DEFENCE ORGANISATION

S&T Program

Chief Defence Scientist 10pt)	
FAS Science Policy)	1 shared copy
AS Science Industry External Relations)	
AS Science Corporate Management)	
Counsellor, Defence Science, London		Doc Control sheet
Counsellor, Defence Science, Washington		1
Senior Defence Scientific Adviser)	1 shared copy
Scientific Adviser - Policy and Command)	
Director General Scientific and Technical Analysis		1
Navy Scientific Adviser		3 copies of Doc Control sheet and 1 distribution list
Scientific Adviser - Army		Doc Control sheet and 1 distribution list
Air Force Scientific Adviser		1
Director Trials		1
Director, Aeronautical & Maritime Research Laboratory		1

Electronics and Surveillance Research Laboratory

Director Electronics and Surveillance Research Laboratory	1
Chief Information Technology Division	1
Chief Electronic Warfare Division	Doc Control sheet
Chief Guided Weapons Division	Doc Control sheet
Chief Communications Division	1
Chief Land, Space and Optoelectronics Division	Doc Control sheet
Chief High Frequency Radar Division	Doc Control sheet
Chief Microwave Radar Division	Doc Control sheet
Research Leader Command & Control and Intelligence Systems	1
Research Leader Military Computing Systems	1
Research Leader Command, Control and Communications	1
Research Leader Military Information Networks	1
Research Leader Secure Communications	1
Executive Officer, Information Technology Division	Doc Control sheet
Head, Information Architectures Group	1
Head, C3I Systems Engineering Group	Doc Control sheet
Head, Information Warfare Studies Group	Doc Control sheet

DSTO-RR-0086

Head, Software Engineering Group	Doc Control sheet
Head, Trusted Computer Systems Group	1
Head, Advanced Computer Capabilities Group	Doc Control sheet
Head, Systems Simulation and Assessment Group	Doc Control sheet
Head, Intelligence Systems Group	Doc Control sheet
Head, Command Support Systems Group	1
Head, C3I Operational Analysis Group	Doc Control sheet
Head, Information Management and Fusion Group	Doc Control sheet
Head, Human Systems Integration Group	Doc Control sheet
Head Intelligence Systems Group	Doc Control sheet
Head Crypto Mathematics Research Group	1
Author (M.K.F. Lai)	6
Publications and Publicity Officer, ITD	1

DSTO Library

Library Fishermens Bend	1
Library Maribyrnong	1
Library DSTOS	2
Library, MOD, Pyrmont	Doc Control sheet

Forces Executive

Director General Force Development (Joint)	1
Director General Force Development (Land)	1
Director General Force Development (Air)	1
Director General Force Development (Sea)	1
Director General Joint Communications and Electronics	1
Director General Management Information and Plans (M-SB-43)	1
Deputy Director Network System Development	1
Deputy Director Electronic Warfare Operations	1
DDCE (M-SB-52)	1
SO INFOSEC (M-SB-38)	1

Army

ABCA Office, G-1-34, Russell Offices, Canberra	4
--	---

Acquisition and Logistics

Director General Information Management and Communications Engineering	1
Assistant Secretary Joint Projects Management	1
DJCPS (APW2-4-10)	1
Director Communications Engineering Development	1
Deputy Director DMDS (APW2-4-20)	1
Deputy Director JP2049 (APW2-4-19)	1

S&I Program

Defence Intelligence Organisation	1
Assistant Secretary Information Security	1
QLL1 (J. Burgess) (M-3-58)	1
Library, Defence Signals Directorate	Doc Control sheet

B&M Program (libraries)

OIC TRS, Defence Central Library	1
Officer in Charge, Document Exchange Centre (DEC),	1
DEC requires the following copies of public release reports to meet exchange agreements under their management:	
US Defence Technical Information Center,	2
UK Defence Research Information Centre,	2
Canada Defence Scientific Information Service,	1
NZ Defence Information Centre,	1
National Library of Australia,	1

Universities and Colleges

Australian Defence Force Academy	1
Library	1
Head of Aerospace and Mechanical Engineering	1
Senior Librarian, Hargrave Library, Monash University	1
Librarian, Flinders University	1

Other Organisations

NASA (Canberra)	1
AGPS	1
State Library of South Australia	1
Parliamentary Library, South Australia	1

OUTSIDE AUSTRALIA**Abstracting and Information Organisations**

INSPEC: Acquisitions Section Institution of Electrical Engineers	1
Library, Chemical Abstracts Reference Service	1
Engineering Societies Library, US	1
American Society for Metals	1
Documents Librarian, The Center for Research Libraries, US	1

Information Exchange Agreement Partners

Acquisitions Unit, Science Reference and Information Service, UK	1
Library - Exchange Desk, National Institute of Standards and Technology, US	1

SPARES 10

Total number of copies: 87

Page classification: UNCLASSIFIED

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION DOCUMENT CONTROL DATA									
				1. PRIVACY MARKING/CAVEAT (OF DOCUMENT)					
2. TITLE Common Security Protocol Security Labelling and its Applications			3. SECURITY CLASSIFICATION (FOR UNCLASSIFIED REPORTS THAT ARE LIMITED RELEASE USE (L) NEXT TO DOCUMENT CLASSIFICATION) Document (U) Title (U) Abstract (U)						
4. AUTHOR(S) M.K.F. Lai, J. Burgess, K. Forrest, H. Daniel & N.F. Parker			5. CORPORATE AUTHOR Electronics and Surveillance Research Laboratory PO Box 1500 Salisbury SA 5108						
6a. DSTO NUMBER DSTO-RR-0086		6b. AR NUMBER AR-009-739		6c. TYPE OF REPORT Research Report		7. DOCUMENT DATE July 1996			
8. FILE NUMBER N9505/010/0112		9. TASK NUMBER ADF93/256		10. TASK SPONSOR HQADF		11. NO. OF PAGES 48		12. NO. OF REFERENCES 12	
13. DOWNGRADING/DELIMITING INSTRUCTIONS N/A				14. RELEASE AUTHORITY Chief, Information Technology Division					
15. SECONDARY RELEASE STATEMENT OF THIS DOCUMENT Approved for public release OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DOCUMENT EXCHANGE CENTRE, DIS NETWORK OFFICE, DEPT OF DEFENCE, CAMPBELL PARK OFFICES, CANBERRA ACT 2600									
16. DELIBERATE ANNOUNCEMENT No limitations									
17. CASUAL ANNOUNCEMENT Yes									
18. DEFTEST DESCRIPTORS Computer Security Secure communication Military communications									
19. ABSTRACT This paper is part of the documentation series produced under the HQADF sponsored task "D6: A Security Architecture for Large, Distributed Multimedia Systems". It shows that the functionality of the Defence adopted Common Security Protocol currently is insufficient to realise an electronic analogue to the paper based formal military correspondence composition. Some minimal structural changes to the protocol data unit and the corresponding procedural changes to the protocol handling are proposed to address the deficiency.									

Page classification: UNCLASSIFIED